



Technical Feasibility Study

for an

On-line Financial Downloading System

Prepared for

**U.S. Department of Labor
Employment Standards Administration
Office of Labor-Management Standards**

Prepared by



SRA International

October 25, 2002
Contract Number: 263-01-D-0050
Delivery Order: J-9-M-2-0021
Task Number: 1
Document Number: DOLOFDS-02D1.0



Version History

| Version | Date | Authors | Change Description | Reviewers |
|----------------|-------------|---|---------------------------|-------------------------------|
| 1.0 | 10.25.02 | S. Averbach, J. Brown, V. Chalasani, P. Halverson | Initial document | L. McCray, J. Smith, M. Yeung |
| | | | | |
| | | | | |

Table of Contents

| | | |
|----------|---|------------|
| 1 | Technical Feasibility Study Overview..... | 1-1 |
| 1.1 | Summary | 1-1 |
| 1.2 | Background | 1-1 |
| 1.2.1 | Current Electronic Filing Process | 1-3 |
| 1.2.2 | Current Environment: Issues and Opportunities..... | 1-4 |
| 1.2.3 | System Objectives and Benefits | 1-4 |
| 1.3 | Assumptions..... | 1-5 |
| 2 | On-Line Financial Downloading System Concept | 2-1 |
| 2.1 | High Level System Requirements..... | 2-1 |
| 2.2 | High Level System Architecture..... | 2-2 |
| 3 | Front End Subsystem | 3-1 |
| 3.1 | Data Import Subsystem | 3-1 |
| 3.1.1 | File Connector | 3-3 |
| 3.1.2 | Direct Connector | 3-4 |
| 3.2 | Data Preparation Subsystem | 3-4 |
| 3.2.1 | Desktop Forms Application | 3-5 |
| 3.2.2 | On-line Web Application | 3-7 |
| 3.3 | Form Transmittal | 3-7 |
| 3.3.1 | Integrated, Web Services-Based | 3-8 |
| 3.3.2 | Integrated, Non-Web Services-Based | 3-8 |
| 3.3.3 | Integrated, Direct Interface | 3-8 |
| 3.3.4 | Standalone Web Application..... | 3-9 |
| 3.3.5 | Email-Based..... | 3-9 |
| 3.4 | Loads and Sizing Estimates..... | 3-9 |
| 3.4.1 | Exported Data File Size | 3-9 |
| 3.4.2 | Front End Data Transmission Bandwidth | 3-12 |
| 4 | Back End Subsystem | 4-1 |
| 4.1 | Processing Subsystem | 4-1 |
| 4.1.1 | Immediate Processing | 4-1 |
| 4.1.2 | Deferred Processing | 4-2 |
| 4.1.3 | Processing Components..... | 4-3 |
| 4.1.4 | Service Components | 4-3 |

| | | |
|----------|--|------------|
| 4.1.5 | Monitoring and Administration | 4-4 |
| 4.2 | Storage Subsystem..... | 4-4 |
| 4.2.1 | Relational Database With XML Extensions | 4-4 |
| 4.2.2 | Native XML Database | 4-5 |
| 4.3 | Query Subsystem..... | 4-5 |
| 4.4 | Loads and Sizing Estimates..... | 4-7 |
| 4.5 | Web Access of Data (Bandwidth Sizing) | 4-8 |
| 5 | Technical Feasibility Analysis..... | 5-1 |
| 5.1 | Forms Preparation | 5-1 |
| 5.2 | Data Import | 5-2 |
| 5.3 | Forms Transmittal | 5-3 |
| 5.4 | Forms Processing | 5-5 |
| 5.5 | Forms Storage | 5-6 |
| 5.6 | Web Query | 5-6 |
| 5.7 | Implementation Technology..... | 5-6 |
| 5.7.1 | Front End Application | 5-7 |
| 5.7.1.1 | PureEdge Internet Commerce System (ICS)..... | 5-7 |
| 5.7.1.2 | Adobe Accelio..... | 5-8 |
| 5.7.2 | Back End Server Platform | 5-8 |
| 5.7.2.1 | J2EE | 5-8 |
| 5.7.2.2 | Microsoft .NET..... | 5-9 |
| 5.7.2.3 | CORBA..... | 5-9 |
| 5.7.3 | Database..... | 5-9 |
| 5.8 | Operational Feasibility | 5-10 |
| 5.9 | Technical Recommendation | 5-10 |
| 6 | DOL Costs Estimate..... | 6-1 |
| 6.1 | Non-recurring Costs for System Development and Deployment..... | 6-1 |
| 6.1.1 | Methodology | 6-1 |
| 6.1.2 | Estimated Non-recurring Costs..... | 6-3 |
| 6.2 | Annual Operating Costs..... | 6-4 |
| 7 | Bookkeeping/Financial Accounting Software Survey..... | 7-1 |
| 7.1 | Objective | 7-1 |
| 7.2 | Methodology..... | 7-1 |
| 7.3 | Survey Results..... | 7-2 |



| | | |
|----------|--|------------|
| 7.4 | Survey Analysis..... | 7-6 |
| 8 | Preliminary Estimate of Burden Change on Filers | 8-1 |
| 8.1 | Summary | 8-1 |
| 8.2 | Assumptions, Methodology, and Approach | 8-1 |
| 8.2.1 | Overall Assumptions | 8-1 |
| 8.2.2 | Overall Methodology | 8-2 |
| 8.2.2.1 | LM-2 Changes Methodology Summary | 8-3 |
| 8.2.2.2 | Form T-1 Methodology Summary | 8-3 |
| 8.3 | Analysis – Summary of Proposed LM-2 Changes | 8-4 |
| 8.4 | Analysis: Non-recurring Incremental Burden on Filers | 8-4 |
| 8.4.1 | Non-Recurring Burden Assumptions | 8-4 |
| 8.4.2 | LM-2 Non-Recurring Burden Methodology | 8-5 |
| 8.4.3 | Form T-1 Non-Recurring Burden Methodology | 8-6 |
| 8.5 | Analysis: Recurring Incremental Burden on Filers..... | 8-7 |
| 8.5.1 | Recurring Burden Assumptions | 8-7 |
| 8.5.2 | LM-2 Recurring Incremental Burden Methodology..... | 8-7 |
| 8.5.3 | Form T-1 Recurring Incremental Burden Methodology | 8-9 |



List of Tables

| | |
|--|------|
| Table 1.1 Annual Financial Report Forms | 1-2 |
| Table 1.2 System Improvement Objective and Strategies | 1-5 |
| Table 1.3 High Level Feasibility Study Assumptions..... | 1-5 |
| Table 2.1 On-Line Financial Downloading System High-Level Requirements..... | 2-1 |
| Table 3.1: Size Estimates for Schedules 1-12..... | 3-10 |
| Table 6.1: Cost Element Structure and Definitions | 6-1 |
| Table 6.2: Average Labor Rates (in Dollars) per Labor Category | 6-2 |
| Table 6.3: Average Labor Rate and Estimated Development Costs | 6-3 |
| Table 6.4: Estimated Hardware and Software Costs..... | 6-3 |
| Table 6.5: Total Development and Deployment Costs | 6-4 |
| Table 6.6: Cost Element Structure and Definitions (Annual Costs) | 6-4 |
| Table 6.7: Operations and Maintenance Tasks | 6-4 |
| Table 6.8: Labor Estimates | 6-5 |
| Table 6.9: Labor Costs for Operation and Maintenance Tasks | 6-5 |
| Table 6.10: Annual Operations and Maintenance Costs Estimates (With Inflation) | 6-6 |
| Table 7.1: Packages Excluded From Survey | 7-2 |
| Table 8.1: Summary of Preliminary Estimate of Burden on Filers..... | 8-1 |

List of Figures

| | |
|---|-----|
| Figure 1.1 Financial Reports Data Search System Sample Report | 1-2 |
| Figure 1.2: Existing Electronic Filing System Process Flow | 1-3 |
| Figure 2.1: High Level System Architecture | 2-2 |
| Figure 3.1: Front End Activities | 3-1 |
| Figure 3.2: Importing Financial Data | 3-2 |
| Figure 3.3: Data Import Components | 3-3 |
| Figure 3.4: Quicken Data Export | 3-4 |
| Figure 3.5: COTS Application | 3-6 |
| Figure 3.6: Web-Based Forms Publishing | 3-6 |
| Figure 3.7: On-line Preparation Application | 3-7 |
| Figure 3.8: Web Service Interface | 3-8 |
| Figure 3.9: Standalone Web Application | 3-9 |
| Figure 3.10: Email-Based Forms Submission | 3-9 |
| Figure 4.1: Immediate Processing Components | 4-2 |
| Figure 4.2: Deferred Processing Components | 4-2 |
| Figure 4.3: Processing Workflow Components | 4-3 |
| Figure 4.4: Query Subsystem | 4-6 |

1 Technical Feasibility Study Overview

1.1 Summary

The Office of Labor-Management Standards is located in the U.S. Department of Labor (DOL), Employment Standards Administration (ESA).

Approximately 6,000 organizations file an LM-2.

The Office of Labor-Management Standards (OLMS) has begun a significant effort to improve the labor organization financial reporting and disclosure process. To increase the transparency and accountability of labor organizations to their members, the reporting improvement effort focuses on:

- Revising the forms on which labor organizations provide information to support more detailed and robust financial reporting
- Refining the method by which labor organizations can electronically prepare and transmit the information
- Enhancing the public's – especially labor union members – ability to access, search, view, and understand the information

OLMS is drafting a Notice of Proposed Rule-Making to revise Form LM-2, which labor organizations with total annual receipts of \$200,000 or more use to file their annual financial reports. OLMS has partnered with SRA International (SRA) to assess the technical feasibility of developing an on-line system that labor organizations can use to complete, electronically “sign,” and transmit information, that labor union members, other government agencies, and the general public can use to access information.

SRA prepared a high-level feasibility overview to assess currently available technology's ability to support OLMS' efforts. SRA concluded that the technology to develop an enhanced LM reporting system exists.

After exploring the technical feasibility further, SRA prepared this document to:

- Define in greater detail a system architecture
- Identify alternatives for developing the system architecture
- Estimate the costs DOL would incur to develop and maintain the system
- Determine the burden that proposed changes would impose on LM filers

1.2 Background

LMRDA, Section 201 requires labor organizations to file annual financial reports with the Department of Labor; Section 208 gives the DOL authority to issue regulations prescribing report forms.

Reporting guidelines are defined in 29 CFR Part 403.

The Office of Labor-Management Standards administers and enforces most provisions of the Labor-Management Reporting and Disclosure Act of 1959 (LMRDA). The Civil Service Reform Act (CSRA) requires most Federal sector labor organizations to file the labor organization reports.

By requiring labor organizations to disclose financial information, the LMRDA benefits the labor-management process and helps ensure labor organizations' fiscal integrity. OLMS also issues, amends, and rescinds rules and regulations prescribing the form and publication of reports that the LMRDA requires labor organizations to file. Table 1.1 lists the labor organization annual financial report forms that current regulations prescribe.



Table 1.1 Annual Financial Report Forms

| Form | Organizations Required to File Form |
|------|---|
| LM-2 | Each reporting union with total annual receipts of \$200,000 or more, and by the parent union for subordinate unions under trusteeship. |
| LM-3 | Each reporting union with total annual receipts of less than \$200,000 may use the LM-3 if not in trusteeship. |
| LM-4 | Each reporting union with total annual receipts of less than \$10,000 may use the LM-4 if not in trusteeship. |

While labor unions' functions and operations have evolved, the financial report forms unions submit today are substantially the same as those they submitted forty years ago. Further, technology has increased the accuracy and efficiency of financial accounting and reporting. Over the last few years, technology – specifically, the Internet – has altered significantly the means by which an organization can share information with its stakeholders.

OLMS has begun to harness the benefits technology offers. In January 2002, after a multi-year development effort, OLMS began distributing to unions software that enables them to complete required reports electronically. Since July 2002, union officers have been able to sign and submit their reports electronically using digital signatures.

Prior to electronic forms' availability, the public could review the reports only at OLMS public disclosure rooms at DOL in Washington, DC, or DOL field offices. Using the DOL's Internet Public Disclosure Room, the public can search for reports (from 2000 and after) using a variety of criteria, including union name, file number, affiliation, and designation name and number. After individuals access a report, they can view and print the report.

The Internet Public Disclosure Room also provides access to The Union Annual Financial Reports Data Search System. This system enables individuals to generate an on-line report displaying selected key information and financial data from union annual reports. For example, a user can indicate she wants to see a report that lists amounts disbursed to officers from each international/national union whose total disbursements were greater than \$1 million. Figure 1.1 displays part of the report this search generates.

Figure 1.1 Financial Reports Data Search System Sample Report

| DATE: 2002-08-20 TIME: 09:58:36 | | UNION LISTING--DISBURSEMENTS LATEST REPORT AVAILABLE | |
|------------------------------------|---|---|------------------------|
| LM # FORM YR END | AFFILIATION NAME DESIGNATION NAME/NUMBER/UNIT NAME CHARTER CITY/STATE | TO OFFICERS | TOTAL DISBURSEMENTS |
| 106 LM-2 2001-12-31 | AFL-CIO NHQ WASHINGTON DC | 443486 | 164246963 |
| 179 LM-2 2000-12-31 | AIR LINE PILOTS ASN AFL-CIO NHQ WASHINGTON DC | 855026 | 190593927 |

1.2.1 Current Electronic Filing Process

Figure 1.2 depicts the existing electronic filing system process flow.

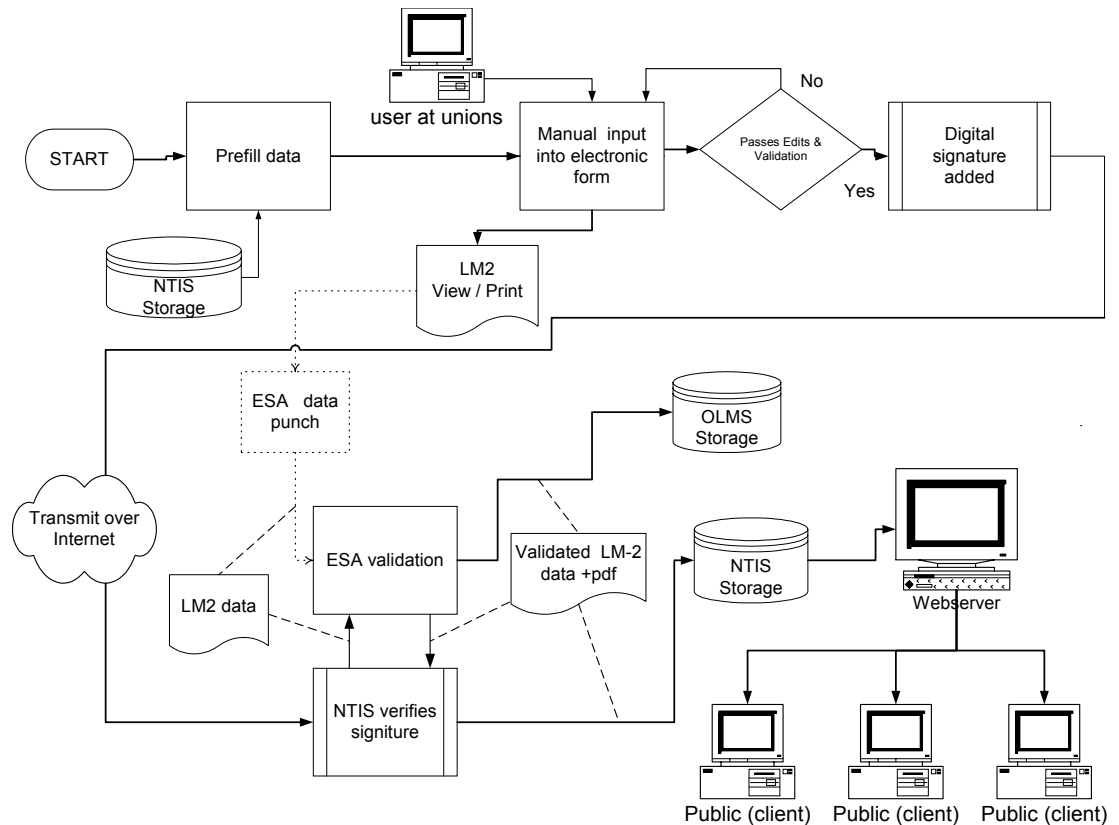
The software will bring required reporting into the 21st century and should save unions valuable time and enhance the accuracy of their financial reports.

– Don Todd, Deputy

In 1998, Congress mandated that DOL develop a system that provides an indexable and searchable database of unions' financial reports, and allows public access to the information.

The Internet Public Disclosure Room's Internet address is www.union-reports.dol.gov

Figure 1.2: Existing Electronic Filing System Process Flow



Currently, about 40 percent of all filers use the electronic filing system to prepare their reports. The other filers complete the forms manually. None of the unions has used the digital signature option.

As part of the annual report package OLMS sends to labor unions, OLMS has been including CDs containing the electronic financial report forms. The labor unions install the CD in their computing environment. The computing environment does not need to have an Internet connection, but an Internet connection offers report preparation benefits.

Using the Internet, the report preparer can connect to the OLMS database, supply the organization's six-digit file number, and have the software "pre-fill" the form with organization-specific information. The preparer enters other information, including financial data, manually.

After the preparer completes the form, the software performs extensive data validation and error-checking. For example, the software performs a “math check” on the amounts entered in the supporting schedules, such as the disbursements schedule. After the data passes validation, the preparer can print the form, have the appropriate officers sign the form, and mail the form to DOL.

When DOL receives the form, staff members enter data manually into an OLMS database. (The process that is specific to only capturing data from the paper forms is shown as dotted lines in Figure 1.2). The report is sent for data extraction and scanning to create a Portable Document Format (PDF) file. OLMS passes the data and the PDF file to the National Technical Information Service (NTIS). NTIS records the data in a database and makes it available to the public on the Internet.

The system is capable of capturing the data from the forms electronically (for unions that submit forms electronically) and transmitting it directly to NTIS. In turn, NTIS passes the data to OLMS. OLMS performs batch data validation, and “throws out” documents that have errors. OLMS then

manually resolves the errors. OLMS records valid data in a database and sends a copy of the data to NTIS.

1.2.2 Current Environment: Issues and Opportunities

The current electronic filing system has improved the filing process and public access to labor organizations' financial information. Yet, OLMS has identified opportunities to improve the process and the supporting information technology:

- Because the existing LM financial reports have not changed significantly since the early 1960s, they no longer present a full and accurate financial view of labor organizations' increasingly multi-faceted and complex business activities. Consequently, union members, the primary stakeholders, do not have access to complete and understandable information about their unions' financial transactions.
- Existing regulations do not require labor organizations to submit financial reports electronically. In the current environment, an LM-2 filer, for example, can complete the LM-2 form by manually entering data into the form. The filer can then print out the form, sign it, and send it to OLMS.
- The current software is designed around Shana Informed Filler. This Commercial Off-The-Shelf (COTS) package has substantial difficulties scaling to accommodate all the data required by the current LM-2, particularly for large unions. OLMS believes that the Informed Filler software may not be able to capture a revised LM-2 form's data.

The anticipated increase in reporting data volume – especially for larger unions – would place a very large burden on the unions who enter data manually. An interface between a labor organization's financial accounting software and the software used to prepare the revised LM-2 would provide a seamless transfer of data and substantially decrease this burden in the long run.

- To make reports available for on-line viewing, OLMS must scan each report that is filed in paper format. To make data available to the Data Search System, OLMS must enter information from the forms into a database. This task is labor-intensive and, therefore, expensive.

1.2.3 System Objectives and Benefits

OLMS seeks to enhance labor organization financial reporting and disclosure in a complex business and sophisticated technological environment. Table 1.3 lists the effort's primary objective, strategies, and tactics.

Table 1.3 System Improvement Objective and Strategies

| | | |
|-------------------|--|---|
| Objective | Provide labor organization members complete, understandable information about their unions' financial transactions, investments, and solvency to place members in a better position than they are today to protect their personal financial interests. | |
| Strategy 1 | Revise Form LM-2 to capture more information. | Strategy 2 |
| | | Require labor organizations to file their reports electronically. |



| Tactics | Tactic |
|---|--|
| <ul style="list-style-type: none">Require disbursements and receipts not otherwise identified be reported in specific categories that provide union members with more detailed information about their unions' activities.Require LM-2 filers to identify individuals and entities who receive major disbursements, and from whom they receive major receipts. | <ul style="list-style-type: none">Realize the benefits of recent technological advances by developing an On-Line Financial Downloading System. Labor organizations would use the system to record and capture organization and financial information, and transmit the information electronically to DOL. The public would be able to query and view labor organization information. |

OLMS expects that meeting its objective will result in the following benefits:

- Greater transparency with respect to union activities
- Reduced levels of embezzlement and financial mismanagement
- Targeted investigations that will lead to more precise inquiries and reduce the risk of random audit burden

1.3 Assumptions

Table 1.5 lists the assumptions under which SRA prepared this study.

Table 1.5 High Level Feasibility Study Assumptions

| ID | Assumption |
|-----|---|
| A-1 | This study focuses only on the technical feasibility of developing an On-Line Financial Downloading System. The study does not examine public policy issues and implications. |
| A-2 | OLMS must obtain regulatory approval required to implement changes to financial reporting forms and for requiring labor organizations to use the OLFDS. |
| A-3 | Labor unions will make the necessary changes to their accounting system to capture the additional details required. |

2 On-Line Financial Downloading System Concept

To assess the technical feasibility of developing the OLFDS, SRA developed a system concept. The two-step concept development approach that SRA followed was:

1. Identify high-level system requirements.
2. Design a high-level system architecture that meets those requirements.

2.1 High Level System Requirements

To help labor organizations meet the proposed requirement to file reports electronically, and to provide additional information on-line, DOL wants to enhance the electronic filing program by developing an On-Line Financial Downloading System. Table 2.1 lists the OLFDS high-level requirements.

Table 2.1 On-Line Financial Downloading System High-Level Requirements

| ID | Requirement |
|----------|---|
| HLR-1.0 | Accept manually entered data in a form-like environment. |
| HLR-2.0 | Import specified financial data pertinent to LM forms from financial packages used by labor unions. |
| HLR-3.0 | Pre-fill data, such as identification details, from an OLMS database. |
| HLR-4.0 | Enable user to view, edit, save, and print LM reports. |
| HLR-5.0 | Perform client side validations on the data, preserving all current edits and validation functionality. |
| HLR-6.0 | Add at least 2 electronic signatures and transmit the report with signature securely over the Internet. |
| HLR-7.0 | Check for digital signature after receipt of data at DOL. |
| HLR-8.0 | Validate incoming data using data in a DOL database and generate an error report if the data does not pass validation. |
| HLR-9.0 | Populate a database with validated data. |
| HLR-10.0 | Provide access to the public over the Internet to individual LM reports down to the detail level. |
| HLR-11.0 | Enable the public's ability to search LM report data over the Internet using, at a minimum, the eight standard reports. |
| HLR-12.0 | The system will be capable of being expanded to include additional reports based on new schedules being captured in revised LM-2 reports. |
| HLR-13.0 | Link to other government entities as needed. |

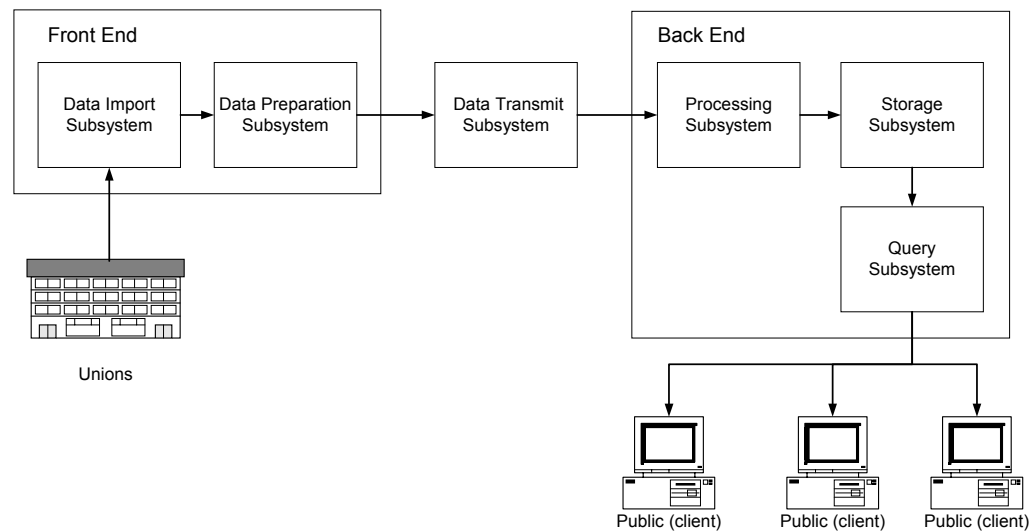
SRA developed the requirements list based on discussions with OLMS and reference documents that OLMS provided.

2.2 High Level System Architecture

Based on requirements, SRA developed a recommended high-level system architecture comprised of three parts: Front End, Back End, and a Data Transmit Subsystem connecting the two Ends.

Front End processes capture data into a form/report and send the report to DOL. The Data Transmit Subsystem transmits completed LM-2 forms to the Back End. The Back End processes and stores data, and provides on-line search capabilities

Figure 2.1: High Level System Architecture



The Front End is comprised of two subsystems:

- **Data Import** processes extraction of data from labor organizations' financial systems
- **Data Preparation** manages manual input of data and/or editing of data and subsequent validation and attachment of electronic signatures

The Back End is comprised of three subsystems:

- **Data Processing** manages data capture, digital signature verification, and transmission of data to the Storage Subsystem
- **Storage** manages data storage
- **Query** manages user queries over the Internet

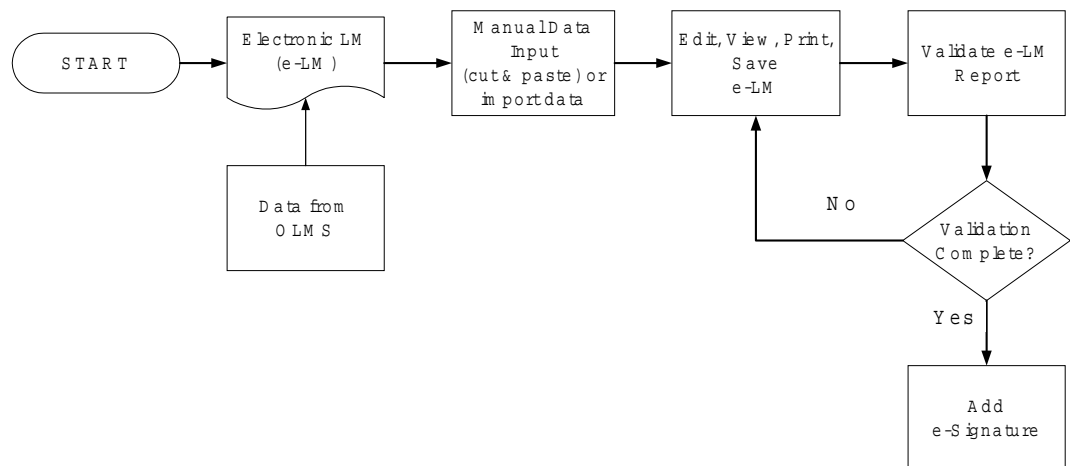
The conceptual break point between the two "Ends" occurs when filers transmit (the Data Transmit Subsystem) the report from the union to DOL. Though a separate subsystem that communicates between the two Ends, the Data Transmit Subsystem is described in the section presenting the Front End.

3 Front End Subsystem

The OLFDS Front End enables union representatives to create, sign, and submit LM-2 forms to DOL for subsequent processing that the Back End performs (see Section 4). Figure 3.1 depicts the following Front End processes:

- DOL sends to the union a diskette or CD (similar to the current method of sending the CD containing Informed Filler), or the union downloads the software from DOL's Internet site
- User enters the file number and, optionally, an OLMS database prefills some fields in the form
- User enters data in some fields
- User invokes import feature or manually cuts and pastes financial data into the form template
- System validates data upon user input
- User can view, edit, save, reopen, and print electronic form template until he or she completes the form
- Two signatories sign report with PKI electronic signature
- Form is submitted to the OLFDS Back End for processing and publishing

Figure 3.1: Front End Activities



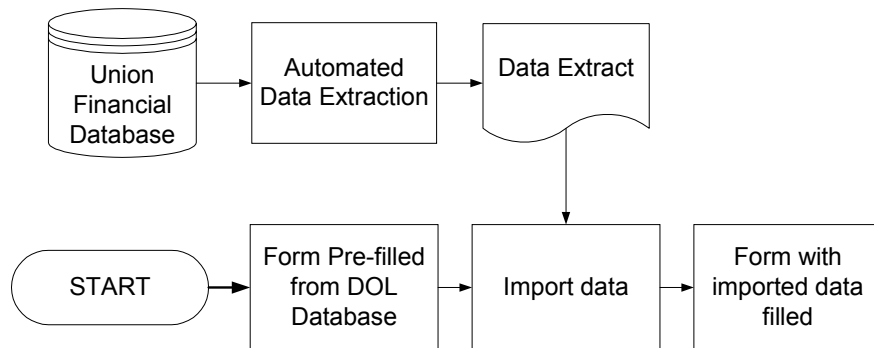
3.1 Data Import Subsystem

The Data Preparation Subsystem's data entry features do not support managing large volumes of data. To handle larger data volume, the Data Import Subsystem interfaces with the union's financial accounting packages to automatically translate union accounting data into the corresponding LM-2 form elements. Labor unions have developed their own custom financial accounting applications or use one of many COTS accounting software packages, such as QuickBooks, Quicken, and Peachtree. Integrating these packages with OLFDS requires developing an interface between the Front End and each financial package.

This is similar to the current system income tax filers use to send tax return data to the IRS. The major financial accounting packages allow data to be extracted, translated, and automatically merged into an electronic tax form which can then be packaged and transmitted over the Internet.

Figure 3.2 presents the Data Import Subsystem functional diagram.

Figure 3.2: Importing Financial Data



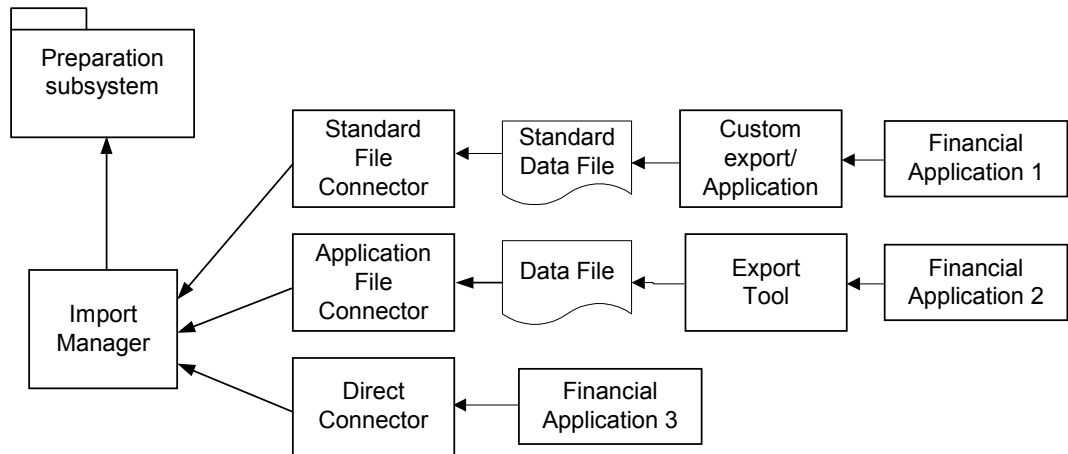
The Data Import Subsystem consists of two main components, as shown in .

- The **Import Manager** integrates Data Import functionality with the Data Preparation Subsystem. It invokes the services of the Application Connector to get the external application data and merges the results with the current report form, mapping from an interchange representation to corresponding LM-2 form items.
- An **Application Connector** extracts data from the external financial applications. It accesses the application, retrieves required data elements, and maps the data to a standard interchange representation.

These two components pass data through a shared interchange representation that captures the financial information in an application-neutral form and allows interchange between applications. XML can be one of the formats in which unions prepare output. XML follows open standards allowing third parties to use it in their products. XML schema defines the information content and format, making data processing, transmission, and storage easier. This also makes transferring data between packages very easy.

The Import Manager is a standard part of the Data Import Subsystem. In most cases, the Application Connector is specific to each financial application; consequently, a connector's design and implementation are highly dependent on the functionality supported by the corresponding application for accessing and extracting data and converting it to the required LM-2 data elements. There are two common approaches for implementing an application connector: using a File Connector and using a Direct Connector.

Figure 3.3: Data Import Components



3.1.1 File Connector

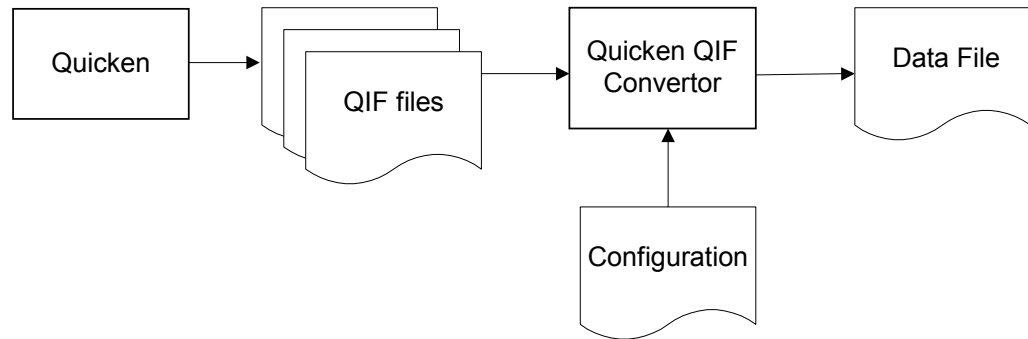
Using the File Connector approach, unions use their system's existing export functionality, a third-party tool, or custom-developed application to create a data file containing the required information. The required connector can be implemented as a standard connector or an application specific connector. DOL can publish a data standard to which the data file should conform. A standard connector would accept such a file directly and no further options except validation would be required. An application specific connector, on the other hand, would perform any additional required processing and prepare the data into a format that can be accepted by the Data Preparation Subsystem.

Figure 3.4 shows an application-specific connector for Quicken. A user would export financial transactions by hand using the Quicken application, creating a set of Quicken Interchange Format (QIF) files, one for each Quicken account. A QIF Converter program would combine all the QIF files into a single OLFDS interchange file, translating each QIF transaction into a corresponding LM-2 transaction. A separate configuration file would be used to define mappings between Quicken category names and the corresponding LM-2 schedule.

It is recommended that DOL consider publishing a data standard and include a Standard File Connector that reads a file formatted to the data standard and passes it back verbatim. This will allow unions to concentrate on acquiring an appropriate export component and on developing an application that can write the file in the standard interchange format, minimizing OLFDS integration costs.

QIF (Quicken Interchange Format) is a specially formatted American Standard Code for Information Interchange (ASCII) text file. It is used to transfer data between different Quicken data files and other programs.

Figure 3.4: Quicken Data Export



3.1.2 Direct Connector

The File Connector approach requires a user to take additional steps to create the required file (running the application or some intermediate application). If a more “seamless” user experience is required, and the union’s financial application supports it, an alternative is to develop a Direct Connector that interfaces directly with the external application without requiring an intermediate file or user intervention to produce a file.

Application Program Interface (API): A set of routines, protocols, and tools for building software applications.

In this model, when the Import Manager invokes the services of the Connector to request financial data, the Connector connects directly to the application, through a standard or vendor-specific (Application Program Interface (API), and requests the required information through this connection. The Connector then reformats the resulting data as needed into the standard OLFDS interchange format and returns it to the Import Manager.

JDBC (Java Database Connectivity): A Java API that enables Java programs to execute SQL statements (requests for information from the database).

If the external application does not support programmatic access, it may still be possible for the Connector to read the application’s data files directly. For example, if the application stores financial data in a relational database, the Connector could use JDBC or ODBC to connect to the database and retrieve the data using SQL. This would create a strong dependency between the connector and the application’s schema definitions, but it allows transparent integration between the Data Preparation Subsystem and the union’s financial data.

ODBC (Open Database Connectivity): A database access method that makes it possible to access any data from any application.

3.2 Data Preparation Subsystem

The Data Preparation Subsystem accepts manually entered data and/or data edits, and it attaches and validates electronic signatures. The user interface is a form in which users enter required data and perform activities required to complete and submit the form. Larger unions can invoke Data Import Subsystem capabilities (see Section 1.1), resulting in transaction data being extracted into the form. Smaller unions could enter financial data manually by cutting and pasting data into a predefined form or template. Unions maintain data in third-party applications, such as Microsoft Excel, from which data can be cut and pasted directly into the application that captures the data.

Additionally, the Data Preparation Subsystem provides viewing, editing, and printing functionality. Because the time needed to complete the form can be extensive, the application can save partially completed forms and restore them during a later session.



Access Certification for Electronic Services (ACES): Built on the Public Key Infrastructure (PKI), ACES is the prime enabler for securing the flow of information on the Internet for the US Government. It ensures privacy, validates identity, and protects information integrity.

Public Key Infrastructure (PKI): A system that verifies and authenticates the validity of each party involved in an Internet transaction.

After the preparer completes the form, the Data Preparation Subsystem allows union officials to sign the information digitally using the current ACES certificates or other Public Key Interface (PKI) credentials as required by DOL. The form can then be transmitted to DOL using the Transmittal features described in Section 3.3.

Two technical approaches that address form preparation requirements are:

- **Desktop Forms Application.** A COTS or custom-developed desktop forms application that supports some form of XML-based forms language (XForms, XFDL, XFA), such as ICS from PureEdge or Accelio from Adobe
- **On-line Web Application.** A web-based service that provides a browser based interface for entering data and importing data from files

3.2.1 Desktop Forms Application

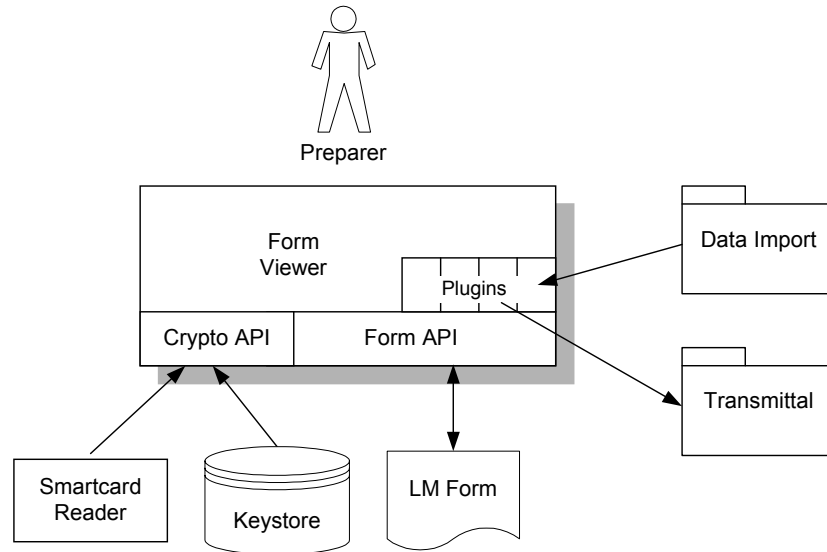
Using a COTS-based form preparation approach, a commercial forms package is the primary component of the Data Preparation Subsystem. Several COTS forms products are available; they typically provide a desktop client application for viewing and filling out forms, an API for extending these applications with custom user interface components, and a set of supporting tools such as forms editors and distribution services. For example, as part of its Internet Commerce System (ICS) forms package, PureEdge includes the ICS Viewer for viewing and completing forms, ICS API for accessing forms data programmatically, ICS Designer for creating forms templates, and the ICS Deployment Server for distributing forms templates and software across the Web. The Accelio Capture suite from Adobe provides similar capabilities.

A key feature of many commercial packages is their ability to support digital signatures. To allow a user to use a public key or other e-Signature mechanism, the application must integrate with the underlying key storage mechanism, such as a browser keystore, a smart card reader, or some other operating-system level cryptographic service. The major forms vendors all provide support for using public key certificates from a variety of vendors and performing the required cryptographic operations to ensure the integrity and authorship of the signed material.

Adopting a COTS forms package for OLDFS would entail the following development tasks:

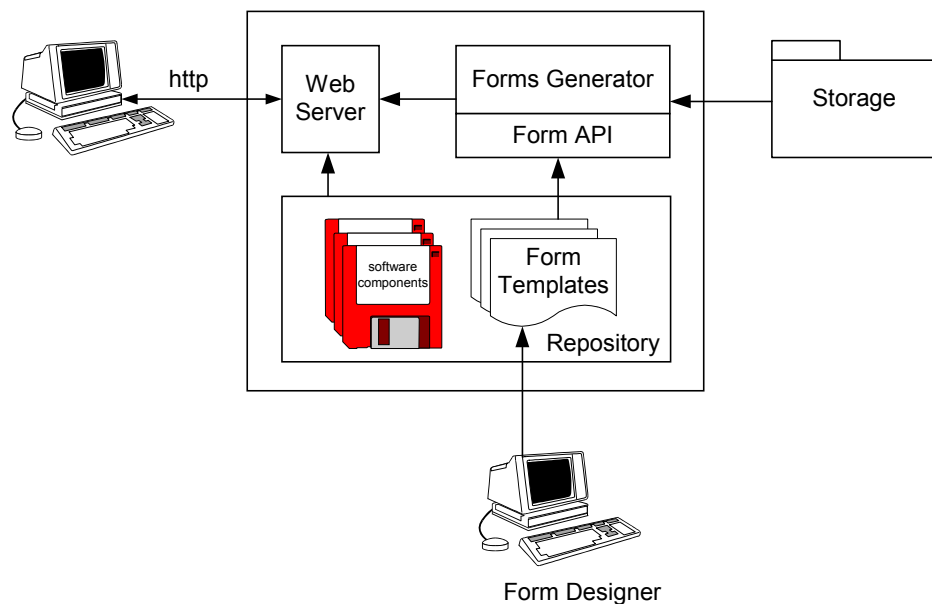
- Using the vendor's form design tools to create electronic equivalents of the LM-2/3/4, and to define data models, presentation items (fields), and any attached constraints or business logic
- Creating additional extensions to integrate with the other OLDFS subsystems

Figure 3.5: COTS Application



- Defining a distribution mechanism, such as a simple web site or distribution CD with a static set of components, to provide union users with software and forms templates to install on their local workstations. Alternatively, the distribution tools could be integrated with the OLFDS Back End to allow forms to be customized for each labor organization. As shown in Figure 3.6, an integrated Forms Generator extension could be used to preload form data, such as union name, address, phone number, and other default information, from a DOL database before sending it to the requesting client.

Figure 3.6: Web-Based Forms Publishing



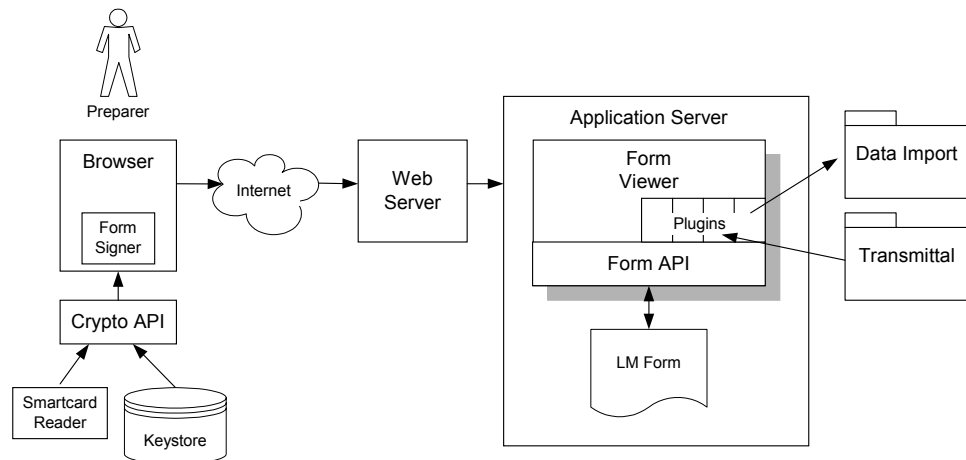
3.2.2 On-line Web Application

An alternative to a desktop COTS-based form preparation approach is an on-line browser-based application for data preparation. Users would access the form on-line through a web browser, requiring a continuous Internet connection while filling the form. Financial data could be imported by uploading a data file, and the server would merge the data into the form. The temporary storage of data would be at the server end. The form can be created through simple web pages or through a COTS package plugged in to the browser. Addition of digital signatures and transmission of the form will be accomplished through the browser. Adopting a browser-based application for OLDFS would entail the following development tasks:

- Designing a web application using custom web pages or using the vendor's form design tools to create electronic equivalents of the LM forms, defining data models, presentation items (fields), and any attached constraints or business logic
- Creating additional extensions at the back end to integrate with the Data Import Subsystem

By hosting the web application in an application server environment, the application could also be integrated with the OLDFS Back End to allow forms to be customized for each labor organization. As shown in Figure 3.7, an integrated Forms Generator extension could preload form data, such as union name, address, phone number, and other default information, from a DOL database before sending it to the requesting client.

Figure 3.7: On-line Preparation Application



3.3 Form Transmittal

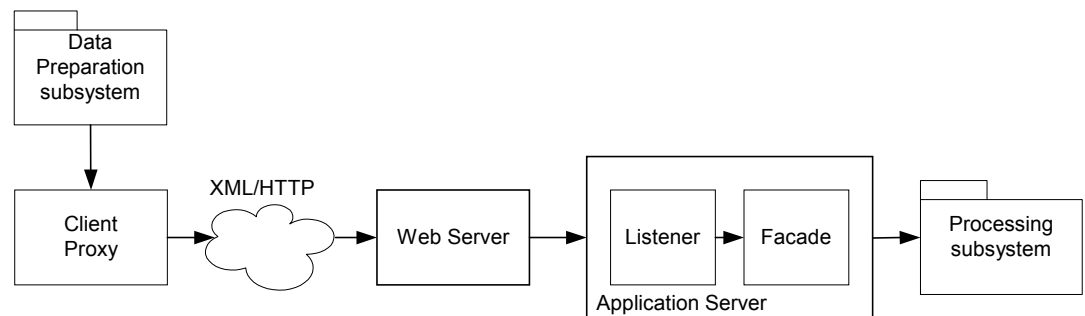
The Data Transmit Subsystem connects the Front End's Data Preparation Subsystem with the Back End Processing Subsystem. The Data Transmit Subsystem

- Transmits signed forms for processing
- Returns acceptance status to user if submission processing is immediate (see Section 4.1.1.)
- Manages receipt for subsequent status query if submission processing is deferred (see Section 4.1.2), allowing the user to monitor submission status (pending, accepted, rejected) as needed

Several alternative designs are feasible. The optimum alternative depends on the degree of integration required.

3.3.1 Integrated, Web Services-Based

Figure 3.11: Web Service Interface



In a web service-based transmittal model, Back End report processing is accessed through a public web service. A web service is a self-contained, self-describing, modular application which can be published, located, and invoked across the web. Web services perform various functions – from simple requests to complicated business processes. Once a web service is deployed, other applications (and other web services) can discover and invoke the deployed service, using widely-accepted, industry-standard protocols (TCP-IP, HTTP, SSL, XML).

***n-Tier Application Architecture* provides a model for developers to create a flexible and reusable application. By breaking up the application into tiers, developers only need to modify or add a specific layer, rather than rewrite the entire application.**

Viewed from an n-tier application architecture perspective, a web service is a veneer for web-based access to functionality implemented by other kinds of middleware. Access consists of service-agnostic request handling (a listener) and a facade that exposes the operations supported by the business logic. The logic itself is implemented by a traditional middleware platform, such as J2EE, .Net, and CORBA.

The Data Preparation Subsystem is extended with a proxy component that encapsulates all interaction with Back End subsystems, hiding the details of the underlying web service protocols. It handles binding and connecting to the remote web service, invoking remote services, and integrating with the rest of the Data Preparation Subsystem. From the user's perspective, forms submission is a seamless extension of the local application functionality.

3.3.2 Integrated, Non-Web Services-Based

Alternatively, Back End services could be exposed using some other remote invocation protocol that is supported directly by the underlying platform (Java RMI, Microsoft COM, CORBA IIOP). As in the web-service model, a local proxy component would be used to hide details of the underlying communications protocol from the rest of the Data Preparation Subsystem.

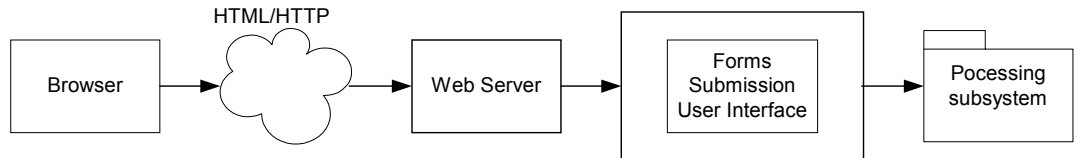
3.3.3 Integrated, Direct Interface

If Data Preparation is offered as a server-hosted web-based application, as described in Section 3.2.2, integration would be similar to the approach described in Section 3.3.1; however, communication between the Data Transmit Subsystem and the Back End would be internal

(possibly even a local method call, depending on how the two are physically deployed), and not subject to the restrictions placed on Internet transactions.

3.3.4 Standalone Web Application

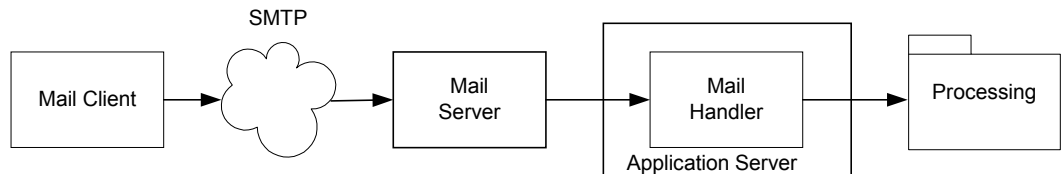
Figure 3.12: Standalone Web Application



An alternative to an integrated Data Transmit Subsystem, based on some form of direct invocation of Back End services through a distributed calling protocol, is a standalone web application that allows user to “manually” upload forms, get a receipt, and check submission status through their browser. This would reduce coupling between the Data Preparation and Data Transmit Subsystems, but at the expense of additional user workload and a less “seamless” experience.

3.3.5 Email-Based

Figure 3.13: Email-Based Forms Submission



Another loosely-coupled transmission approach is to provide an email-based submission interface. In this model, the forms submitter emails the signed form to a designated DOL email address. An automated Mail Handler monitors the associated mailbox, screens non-conformant messages, extracts forms, and submits them to the Processing Subsystem for submission. Receipt is delivered by return mail to the sender’s address.

3.4 Loads and Sizing Estimates

3.4.1 Exported Data File Size

The proposed electronic filing system will accommodate a seamless flow of electronic information from labor unions to the Department of Labor. Beginning with the labor union systems themselves, the data must be pulled out of the financial application unions currently use via an agreed-upon format and structure. This extracted information will then be imported into DOL’s databases where it can be made more easily and directly available to information seekers via the



Internet. The tools that enable the initial process of extracting the necessary information into a specified format make up the Front End of the baseline system.

In creating specifications for such a system, it is vitally important to have a clear understanding of the information that will be handled by the new system. Data types and volume are key design and purchase decision drivers. Physical file size requirements for the OLFDS were extrapolated based on the data types and volume currently being reported to the Department of Labor via the LM-2 form and its associated schedules.

Using the proposed revised LM-2 form, we examined information labor unions are asked to provide. For the purposes of estimating file sizes and application loads, data can be divided into two categories: (1) fixed-size and (2) variable-size information. All LM-2 filers must provide fixed-size, or static, information, such as union name and address; consequently, fixed-size data fields are roughly the same size for all filers. It is estimated that fixed-size fields will require less than 1 kilobyte in physical file memory and storage.

The variable-size portion of the form contains data that is entered on LM-2 Schedules 1-22. Because these schedules report transaction-level data, data volume varies significantly from union to union. Table 3.1 lists estimates for file size per schedule based on "worst-case" assumptions regarding the maximum number of entries on each schedule according to a survey conducted by OLMS. **Note:** The estimates for the accounts receivable and accounts payable aging schedules could not be determined by survey data; we have assumed that they contain 200 entries, which is very conservative. In practice we would expect this number to be smaller.

Table 3.1: Size Estimates for Schedules 1-12

| No | Schedule | Number of entries | Size of each entry (bytes) | Total size (K bytes) |
|----|---|-------------------|----------------------------|----------------------|
| 1 | Accounts receivable aging | 200 | 50 | 10 |
| 2 | Loans receivable | 94 | 150 | 14.1 |
| 3 | Sale of investments and fixed assets | 9 | 50 | 0.45 |
| 4 | Purchase of investments and fixed assets | 20 | 50 | 1 |
| 5 | Investments other than U.S. Treasury securities | 20 | 20 | 0.4 |
| 6 | Fixed assets | 48 | 100 | 4.8 |
| 7 | Other assets | 14 | 100 | 1.4 |
| 8 | Accounts payable aging | 200 | 50 | 10 |
| 9 | Loans payable | 3 | 150 | 0.45 |
| 10 | Other liabilities | 16 | 30 | 0.48 |



| No | Schedule | Number of entries | Size of each entry (bytes) | Total size (K bytes) |
|----|--|-------------------|----------------------------|----------------------|
| 11 | All officers and disbursements to officers | 32 | 100 | 3.2 |
| 12 | Disbursements to employees | 1317 | 100 | 131.7 |
| | | | TOTAL: | 157.98 |

Based on these estimates and calculations, the file size for data from Schedules 1-12 would be approximately 158K. As previously estimated, the fixed-size portion of the file would require only 1K. By rounding up to a conservative value of 200K we can account for all but the information from schedules 14-22¹. In an average scenario, the expected size would be 1/100 of this, or about 2K, assuming that the number of entries is proportional to the revenue as an average LM-2 filer has receipts of about \$2.5Million.

The remaining LM-2 schedules are expected to make up the bulk of the data included in each filing, and contain information pertaining to receipts and disbursements reported at the transaction level. For the purposes of estimating file sizes, we have made the following assumptions:

- Each transaction (whether a receipt or a disbursement) is 150 bytes in size. The transactions include the name, address of the vendor, the type of business, purpose of the receipt or disbursement, date, and the amount received or paid.
- Value of each receipt transaction is \$5000. (In practice, reported receipts would be much larger than \$5000.)
- Value of each disbursement is \$1000. (In practice, most reported disbursements would be much larger than \$1000.)
- Approximately 50 percent of the total disbursements fall into categories for which itemized transactions are reported (based on a DOL survey of 200 unions).
- The percentage of the total receipts that fall into categories which are reported on the LM-2 schedules is expected to be small, as the vast majority of receipts come from union membership. As such, those transactions will not be itemized. We shall assume a conservative result in which 50 percent of total receipts will be reported – the same value used for percentage of total disbursements reported. (In practice, we would expect this number to be less than 25 percent.)
- The amounts received and the amounts disbursed are approximately equal over the course of a year.

At the highest end of the scale, the largest unions have approximately \$300 million in receipts per year. Applying the 50 percent receipts reported and \$5000/transaction assumptions, then:

$$0.5 \times (\$300,000,000/\$5000) = \mathbf{30,000 \text{ receipt transactions per year}}$$

Similarly, assuming \$300 million in disbursements, 50 percent of all disbursements are itemized, and each disbursement is \$1000, then:

¹ Schedule 13 contains data that is largely static and consistent in size from one union to the next. As such, it has been included in the calculations for the fixed-size portion of the file.

$$0.5 \times (\$300,000,000/\$1000) = \mathbf{150,000 \text{ disbursement transactions per year}}$$

In practice, the number of reported transactions is expected to be much smaller -- many disbursements will be for amounts much larger than \$1000. In addition, repeated disbursements to the same entity will be combined into one larger entry for reporting purposes and most organizations typically deal with only a limited number of vendors.

Using these estimates of reported transactions to determine file size in MB:

| | | | | | | |
|-------------------------|---|------------------------------|---|-------------------|---|--------|
| 30,000 | + | 150,000 | x | 150 | = | 27 |
| Receipt transactions | | Disbursement transactions | | K/transacti on | | M B |

If this value (27 MB) is combined with the estimates from the fixed- and variable-size estimate (200K, or about 0.2MB), the overall file size for the worst case scenario is approximately 27.2MB.

By rounding the file size up to 30 MB, and assuming that the file containing the extracted LM-2 data is written using the XML file format (this adds additional 50 percent of the file size (15 MB) to account for the overhead associated with the XML tags contained in the file, the maximum size for a single file is:

$$30\text{MB} + 15\text{MB} = \mathbf{45 \text{ MB}}$$

We would expect this number to be much smaller than this in practice. Assuming that most of the itemized receipts range from \$5000 to \$5 million and the itemized disbursements range \$1000 to \$1 million, we can estimate file size as follows:

Because the survey indicates that approximately 50 percent of the disbursements are greater than \$1000, if we assume a normal distribution, we can estimate that the average for the distribution is \$1000. If we consider the upper half of the distribution, or only transactions above \$1000, and assume that transactions of \$1 million or more occur in only about 0.025 percent of the cases, we can estimate the standard deviation for the distribution to be about \$330,000.

If we use a more conservative estimate, or \$200,000 for the standard deviation, we estimate that the average value of disbursement is approximately \$80,000. If we use a more conservative number of \$50,000, the number of disbursements would be 1/50 of what we had estimated under the worst case scenario. Similarly, the expected number of receipt transactions using this analysis would be 1/50 of what we had estimated under the worst case scenario. Using this analysis the file expected size for a single file using our assumptions would be:

$$(27.2/50 + 0.5) \times 1.5 = \mathbf{1.6\text{MB}}$$

3.4.2 Front End Data Transmission Bandwidth

In addition to accommodating the file size of the LM-2 data that has been exported from a union's financial applications, the proposed electronic filing system must also plan for the needs of the unions when it comes time to transmit the forms. Unions have approximately 90 days to submit form LM-2 after the end of their financial years. Two-thirds of LM-2 filers (approximately 4000 unions) operate on a January-December fiscal year. Approximately 1000 unions' fiscal years end in June, and another 1000 unions' fiscal year ends in September. Because the largest number of unions' fiscal year ends in December, we can assume that the peak load would occur during the 90 days following December 31.



Given a maximum data total for all filers of approximately 2025 MB², the average file size is approximately 337KB:

$$2025 \text{ MB} / 6000 \text{ unions} = \mathbf{337\text{KB}}$$

This can be rounded up to 400KB. The time required to transfer 400 KB on a dial up connection (56Kbps) would be:

$$(400\text{KB} \times 8 \text{ bits}) / (56\text{Kbits/second}) = 57.4 \text{ seconds, or approximately } \mathbf{60 \text{ seconds}}$$

Assuming that approximately 25 percent of the unions (1500) try to file on the last date and that the time on that last day when they actually submit their form is distributed evenly throughout the work day, we would expect about 3 users attempting a connection each minute on an average:

$$1500 \text{ submittals} / (8 \text{ hours} \times 60 \text{ min/hr}) = \mathbf{3.125 \text{ submissions/minute}}$$

Assuming that a Poisson distribution³ accurately describes the instances of union filing submissions, creating capacity for 7 users connected simultaneously will ensure a 99.5 percent service level. To support an average of 7 users (dial up) simultaneously, a bandwidth of (56Kbps x 7) or about 390 Kbps is required. Assuming that across the unions there is an even mix of dial up and high speed Internet connections (DSL 360 kbps), DOL should plan for a bandwidth of:

$$(56\text{Kbps} \times 3.5) + (360\text{kbps} \times 3.5) = \mathbf{1.5 \text{ Mbps (approximately)}}$$

² This value for the maximum file size for all LM-2 forms from all unions combined is calculated in detail in Section 4.4.

³ A Poisson distribution is most commonly used to model the number of random occurrences of some phenomenon in a specified unit of space and time. In this case, it models the probability that a certain number of users will be connected at any given time during the last day to file.

4 Back End Subsystem

The Back End captures and validates LM financial reports data submitted from the Front End, transforms report information into persistent storage, and provides public access to the database through an Internet-based search facility. The primary Back End functions are:

- Receive a report from the Data Transmit Subsystem
- Store the report in an intermediate document repository for validation
- Validate report data against DOL business rules, verify data integrity, and check digital signatures
- Identify errors (DOL sends reports containing these errors back to the union. The union corrects the errors and resubmits. Errors are also reported to a summary log for OLMS review.)
- Store accepted report information in a searchable relational database
- Enable Internet access and database search capabilities

The Back End's three subsystems, **Processing**, **Storage**, and **Query**, perform distinct functions and have a well-defined set of interfaces that other subsystems use to request services. All inter-subsystem communication occurs through these interfaces, allowing internal implementation details to be encapsulated.

4.1 Processing Subsystem

The Processing Subsystem:

- Validates data
- Handles errors
- Manages other business-related processing that must be performed before reports are saved to permanent storage and made available to the public

A **Submission Manager** provides the Processing Subsystem's public interface. There are two alternative form processing operation models: **immediate** and **deferred**. The interface and behavior of the Submission Manager (and, therefore, of the Processing Subsystem) depends on which model is adopted.

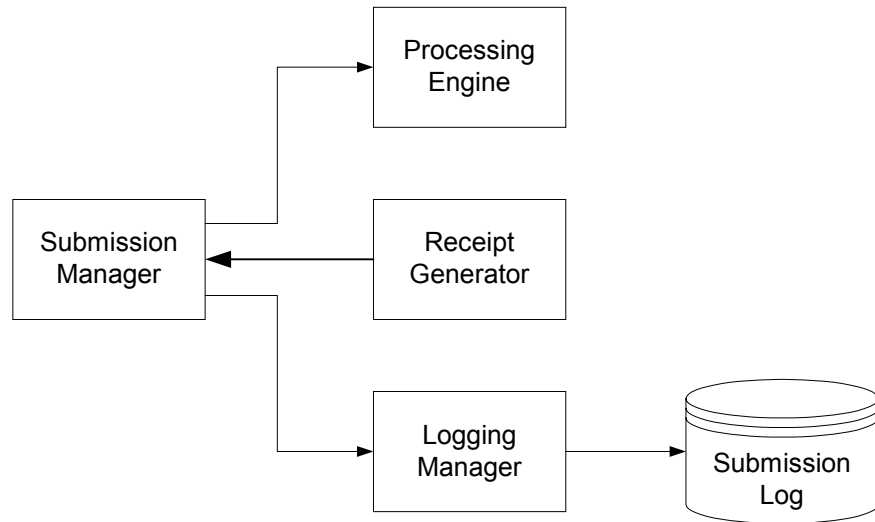
4.1.1 Immediate Processing

In the **immediate** model, depicted in Figure 4.1, a report is processed immediately upon receipt. The caller must wait for validation and storage (if accepted) to complete before continuing. The response from the submit service is either "accepted" or "rejected" (with errors reported back to the requester). Because accepted reports are filed with the Storage Subsystem, no intermediate work queue or associated server storage is required.

In the immediate model, the Processing Subsystem provides a single service:

Submit new form: returns receipt and status (accept/reject).

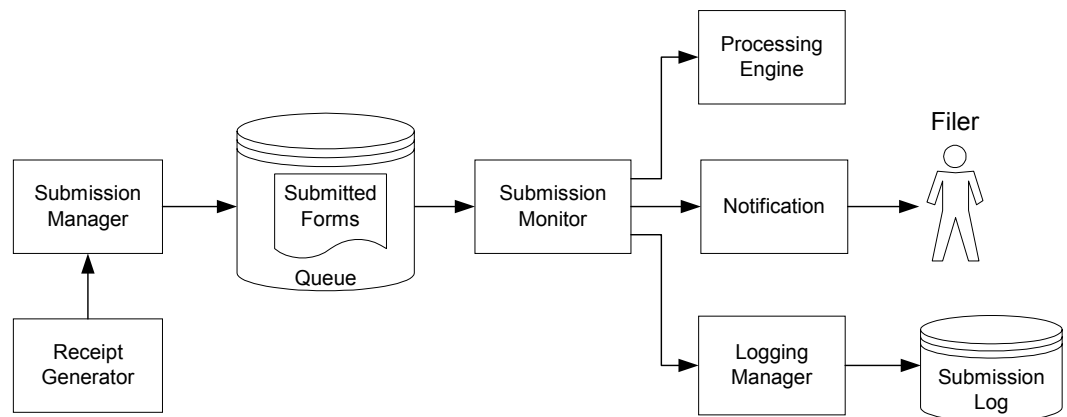
Figure 4.1: Immediate Processing Components



4.1.2 Deferred Processing

With a **deferred** model, represented in Figure 4.2, submitted reports are not processed immediately. Instead, the form is added to an intermediate work list which is monitored by a **Submission Monitor**. This module selects a form and invokes the processing engine on it at appropriate intervals (for instance, as a periodically-recurring task, or when a predefined amount of data has accumulated). Because processing status (accept/reject) can no longer be provided directly to the caller as part of the response, this information is provided separately, either through the notification service (push) and/or by allowing the caller to request status on demand (pull).

Figure 4.2: Deferred Processing Components



To allow the caller to monitor the status of pending submissions and provide proof of submission, a preliminary receipt is generated and returned to the caller immediately. This receipt will be used

by all subsequent interactions between the Data Transmit Subsystem and the Processing Subsystem regarding the submitted report.

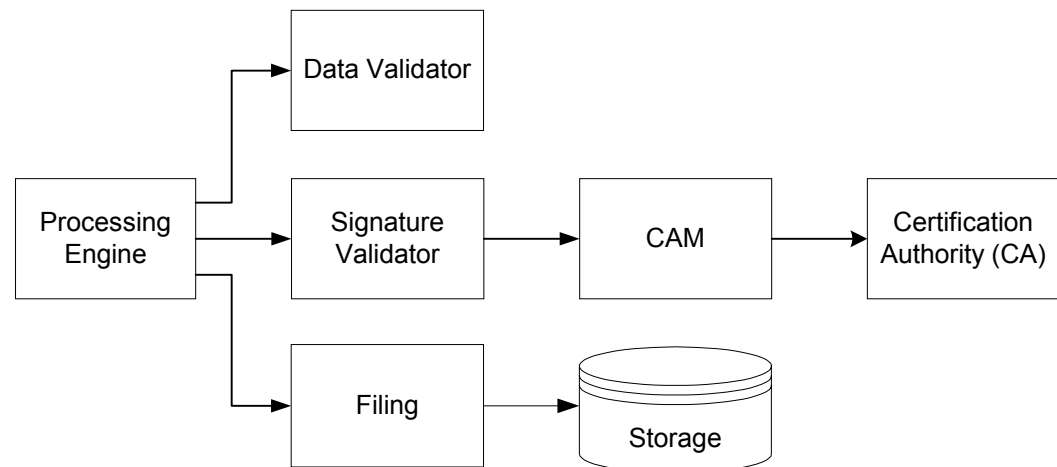
In the deferred model, the Processing Subsystem provides the following services:

- **Submit new form:** returns receipt
- **Check status** of form: returns pending/accept/reject
- **List pending forms** (for use by OLMS staff)

4.1.3 Processing Components

The specific business logic that must run on a submitted form is the same whether processing is immediate or deferred. As depicted in Figure 4.3, the **Processing Engine** processes a form using a number of discrete components, each of which is responsible for a specific step in the processing workflow. As with the Processing Subsystem itself, processing components expose their functionality through well-defined interfaces, allowing them to be replaced or updated independently of each other.

Figure 4.3: Processing Workflow Components



- The **Data Validator** verifies the correctness of submitted data (per OLMS business rules).
- The **Signature Validator** verifies the correctness of the digital signatures. It validates the signing certificates by checking that they have not expired nor been revoked; and it verifies that the signed data has not been altered.
- Certificate validation is delegated to an instance of the **Certificate Arbitrator Module (CAM)**, an application-level router that routes certificates from relying party programs (such as the OLDFS) to the issuing certification authorities (CAs) for validation. By interfacing directly with the CAM, OLDFS can interact seamlessly with multiple CAs – specifically, those CAs issuing ACES certificates to the union signers.
- The **Filing** component saves accepted forms to the Storage Subsystem.

4.1.4 Service Components

Other common services used by various Processing components include (refer to Figure 4.2):

- The **Receipt Generator** is used by both immediate- and deferred-model submission managers to assign a filing number and create an official (digitally-signed) record of receipt.
- The **Notification Service** connects to email agents or other messaging systems to transmit updates of submission status or other information directly to filers.
- The **Logging Manager** maintains a complete record of all submissions and processing steps. This log is used for diagnostic and review purposes, as well as providing the evidentiary basis for OLMS records management practices.

4.1.5 Monitoring and Administration

An administrative “back-office” application which interfaces with internal elements of the Processing Subsystem would allow OLMS staff to monitor submissions, track application statistics, manually update submissions, and otherwise manage the automated submission systems. This application would be most easily provided as a web-based application, using the same infrastructure as the union- and public-facing applications, but restricted to OLMS staff members.

4.2 Storage Subsystem

The Storage Subsystem is responsible for managing all persistent OLFDS application data, specifically the submitted reports and associated filing data. This subsystem supports the following operations

- **Storing** a report (after validation and acceptance processing by the Processing Subsystem)
- **Retrieving** an original report (by an identification number or other criteria) for display
- **Querying** report data for analysis and review

There are two possible approaches for storing and managing application data:

1. Relational database with XML extensions
2. Native XML database

4.2.1 Relational Database With XML Extensions

The most common approach to persistent storage is to use a relational database, which represents application data as rows in data tables. Report data must be extracted from the submitted forms and inserted into one or more tables for subsequent query and retrieval by the Query application. The report forms themselves must also be captured as table values (for example, as a CLOB value), in order to support viewing the original signed document.

Report forms can be represented using XML. Database vendors, such as IBM, Microsoft, Oracle, and Sybase, provide tools and extensions to help automatically convert XML documents into relational tables. Vendors also provide support for maintaining XML documents in their native form. For example, the Oracle XML SQL Utility (XSU) models XML document elements as a collection of nested tables. Enclosed elements are modeled by employing the Oracle Object data type. XSU mapping facilities allow a developer to define how XML document elements are mapped to table columns on insertion. Direct XML document storage is handled by defining an XML Type data type, providing SQL extensions to query the documents using an XPath-based search language.

CLOB (Character Language Object) is used to store large, single-byte character set data.

Binary Language Object (BLOB): A collection of binary data stored as a single entity in a database management systems (DBMS). BLOBs are used primarily to hold multimedia objects such as images, videos, and sound, though they can also be used to store programs or even fragments of code. Not all DBMSs support BLOBs.

Similarly, the IBM DB2 XML Extender allows storing XML documents either as BLOB-like objects or as decomposed into a set of side tables. The latter transformation, known as an XML collection, is defined in XML 1.0 syntax.

Once the XML reports have been converted and stored into relational tables (a process called “shredding”), the data from the tables can be queried using standard SQL-based interfaces. All major database vendors provide a full set of tools for accessing and managing relational data through standardized interfaces (for example, JDBC, .NET, and ADO).

If report forms are represented in some format other than XML, additional transformation components may be required to extract application data elements from submitted reports and insert them into the appropriate tables.

4.2.2 Native XML Database

An alternative to storing OLFDS application data in a relational database is to use a Native XML Database (NXD). An NXD stores XML data and all components of the XML model intact. An NXD has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage. NXDs are based on a document model that is closely aligned with XML or one of XML's related technologies like the Document Object Model (DOM). This model includes arbitrary levels of nesting and complexity, as well as complete support for mixed content and semi-structured data. This model is automatically mapped by the NXD into the underlying storage mechanism. The mapping used will ensure that the XML-specific model of the data is maintained.

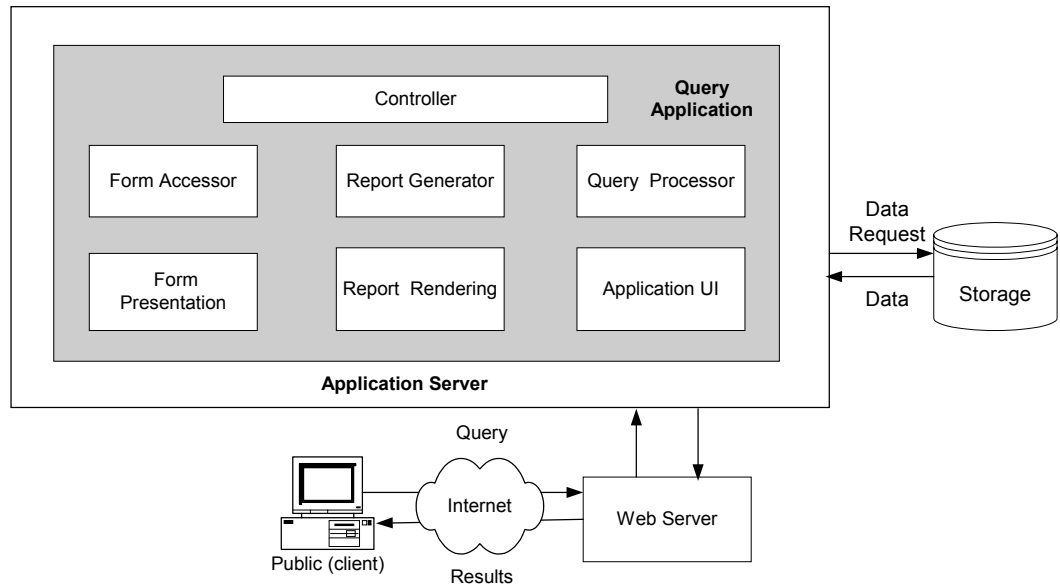
NXDs manage collections of documents, allowing documents to be queried and manipulated as a set. This is very similar to the relational concept of a table. NXDs diverge from the table concept in that not all native XML databases require a schema to be associated with a collection. This means that any XML document can be stored in the collection, regardless of schema. Queries can still be constructed across all documents in the collection. Having schema-independent document collections provides the database a high degree of flexibility and makes application development easier. Unfortunately, it also reduces the ability to manage data integrity.

Documents or document elements are queried using an XML-oriented query language such as XPath or XQuery. These languages allow search criteria and extraction elements to be specified as expressions based on the elements and attributes of the target documents. For example, a query for “Reports from unions with receipts greater than \$100k.” might be represented as an XPath expression of the form “/form[@id='LM-2 and data/item[@id='receipts' and cvalue > 100000]]”. To improve the performance of queries, NXDs support the creation of indices on the data stored in collections. These indices can be used to improve the query execution speed dramatically.

4.3 Query Subsystem

The OLFDS Query Subsystem provides web-based access to the stored union financial information. Specific subsystem functions include processing user requests, retrieving the data from the database, and presenting the results to the user in a readable format. The conceptual design for the Query Subsystem is shown in Figure 4.4.

Figure 4.4: Query Subsystem



The Query Subsystem includes a **Web Server** and an **Application Server**. The Web Server facilitates presentation of an interface to the user for entering the query over the Internet, transmission of query to the application server and presentation back to the user of the results received from the application server. The Application Server would host the actual presentation and business logic of the reporting application.

The reporting application itself is organized based on a conventional Model-View-Controller approach. Separate components are responsible for accessing Back End services, rendering the results for presentation, and coordinating application event flow. This separation facilitates extending the Query application to handle new report types or other functionality with minimal disruption to existing code.

Application user interface components will be responsible for user interaction (displaying forms, dispatching requests). Queries such as those for specific data elements from reports of one or more unions would be handled by a **Query Processor**, whereas the request for an original LM-2 financial report would be handled by a **Form Accessor**. These query components would use the interfaces exposed by the Storage Subsystem.

Additional components would be responsible for presentation of results to the user. The **Form Presentation** component would display an original signed LM-2 form. Extended data reports would be created by the **Report Generator** and **Report Rendering** components, such as those provided in a commercial report generation tool. Many of the application server platforms (particularly those from vendors with a corresponding relational database product) include their own web-based reporting applications, such as Oracle 9iAS Report Services and Microsoft SQL Server's report capabilities. Other options include separate third-party applications, such as Crystal Reports, Cognos, and Web Focus, which utilize ODBC or JDBC connections to access the database. The advantage of using third-party reporting components is that the developer only provides the integration; the report functionality is an out-of-the-box capability. In contrast, built-in capabilities must be further developed into an application.

4.4 Loads and Sizing Estimates

After the relevant data has been extracted from the union's financial applications, it will be transmitted to the Department of Labor. Data must then be imported into and stored in a database. This database must be robust enough to handle the information unions submit. Using assumptions similar to those applied to the Front End data sizing estimates, it is possible to estimate the maximum storage required to hold all data supplied from all reporting labor unions.

The contents of this Back End database will be primarily the data from the LM-2 forms. Given that the total receipts of all LM-2 filers is approximately \$15 billion. SRA estimated Back End storage requirements using the following assumptions:

- Each transaction (whether a receipt or a disbursement) is 150 bytes in size. The transactions include the name, address of the vendor, the type of business, purpose of this receipt or disbursement, date, and the amount received or paid.
- Value of each receipt transaction is \$5000. (In practice, reported receipts would be much larger than \$5000.)
- Value of each disbursement is \$1000. (In practice, most reported disbursements would be much larger than \$1000.)
- Approximately 50 percent of the total disbursements fall into categories for which itemized transactions are reported (based on a DOL survey of 200 unions).
- The percentage of the total receipts that fall into categories which are reported on the LM-2 schedules is expected to be small, as the vast majority of receipts come from union membership. As such, those transactions will not be itemized. We shall assume a conservative result in which 50 percent of total receipts will be reported – the same value used for percentage of total disbursements reported. (In practice, we would expect this number to be less than 25 percent.)
- The amounts received and the amounts disbursed are approximately equal over the course of a year.

Based on these assumptions,

$$0.5 \times (\$15 \times 10^9 / \$5000) = \mathbf{1.5M \text{ receipt transactions}}$$

$$0.5 \times (\$1.5 \times 10^9 / \$1000) = \mathbf{7.5M \text{ disbursement transactions}}$$

Therefore, 9 million transactions are reported.

150 bytes/transaction would require storage for a file size of:

$$9 \times 10^6 \text{ transactions} \times 150 \text{ bytes/transaction} = \mathbf{1350 \text{ MB}} \text{ (approximately)}$$

Assuming that all files will be sent using the XML file format (requiring an additional 50 percent to the file size to account for the overhead associated with the XML tags), the maximum storage requirement is:

$$1350 \text{ MB} + 675 \text{ MB} = \mathbf{2025 \text{ MB}} \text{ (slightly more than 2 GB)}$$

In practice, we would expect this size to be about 1/50 of this, or about 45MB, using analysis similar to that presented in Section 3.4.1.



4.5 Web Access of Data (Bandwidth Sizing)

A goal of the proposed electronic filing system is to provide a means for the public to access the data reported to the Department of Labor from the labor unions via the LM-2 form and associated schedules. This access is currently provided via a public website which enables searching of the LM-2s (currently manually coded into PDF format and uploaded to the site). Usage of the existing site is significant -- over a period of approximately four months (June 4 to September 24, 2002) the site received 170,000 hits, or an average of about 1550 hits/day. Following the implementation of the proposed electronic filing system, substantially more information from the LM-2s will become available. This is expected to significantly increase the load on the website. If we assume that the average amount of data transferred is approximately 10 kb per hit and each hit will require about 10 seconds to service, and further assume that the load on the new website will increase by a factor of four (resulting in about 6,000 hits per day for an eight hour work day), we arrive at the conclusion that a capacity of 5 simultaneous connections to the web server should be able to provide a 99.5 percent level of availability (users will receive their data within the estimated 10 seconds 99.5 percent of the time). The bandwidth required to support this load is about 1 Mbps, assuming an even mix of high speed and dial up connections.

5 Technical Feasibility Analysis

Sections 3 and 4 presented a number of design alternatives for OLFDS subsystems. This section discusses the risks and benefits of each option, and summarizes SRA's recommended approach for each subsystem based on our current understanding of DOL's business goals and functional requirements.

5.1 Forms Preparation

Two possible design alternatives were presented for creating, editing, and completing financial reports: (1) a client-side desktop application based on a COTS forms product, and (2) a browser-based approach in which the application is hosted on an application server. When evaluating the alternatives, consider the following factors:

Technology. Both desktop and web-based approaches are well-established and supported by current technology. Many systems currently in operation use one or more of these approaches, including DOL's current electronic filing system for LM-2, SEC's EDGAR, and the Federal Election Commission's (FEC's) on-line financial downloading system. A number of mature, commercially available products support the required report preparation functionality without further customization (see Section 5.7.1).

User network environment. The web-based solution requires the user to maintain a continuous network connection to create and modify forms. The desktop model, by contrast, only requires a user to be connected when submitting reports or checking acceptance status; all other preparation tasks can be accomplished in "offline" mode.

Application integration. Until web service-based, Internet-wide application integration becomes mature and widespread, data transfer between a web-based preparation environment and a union's financial application is limited to manual upload of an exported data file. This results in a more complicated process for end users. By comparison, a desktop application running on a union client with full network access can potentially be seamlessly integrated with other union business systems using whatever technology or connection mechanism is available. Similarly, a desktop application can directly integrate with local PKI components for signature application and verification, whereas a server-side application must download the necessary client-side objects to perform the same tasks, introducing additional complexity into the solution.

Application complexity. The distributed nature of a web application, which requires the coordination of components residing on web servers, application servers, and local client browsers, may result in a more complex solution than a standalone web application. Application complexity affects development, system administration, and support efforts.

Deployment and updates. In order to access a web-based solution, a union user needs only a web browser; all other components are either deployed and executed on the server, or are downloaded automatically by the application itself. Any changes or updates need only be deployed once to the server, and are immediately made available to the entire user community. However, if the application requires client-side software components, such as ActiveX controls or Java applets, which are excessively large, constant downloading may be prohibitive.

In contrast, a desktop-based application must be distributed to the client machines, either via physical medium, such as a CD-ROM, or electronically via the Internet. Once distributed, the software must be installed and configured correctly. If any updates are made to software

components providing bug fixes or additional functionality, the distribution and installation process must be repeated for every user. Different existing versions of the same application complicate support and development tasks.

Some forms vendors have taken steps to address the desktop deployment issue. PureEdge, for example, provides a server-based software repository that manages the initial download and installation of their forms viewer application, and automatically updates the client as new versions are made available. This same product can also manage the forms templates themselves, avoiding the need to mass-distribute updated DOL forms every year.

5.2 Data Import

The Connector-based design of the Data Import Subsystem supports a range of data entry capabilities, from pure manual cut-and-paste, to file-based data import, to fully integrated connections to financial applications. From a technical design perspective, each of these is possible using the base framework described. Regardless of approach, every distinct financial application will likely require specific software that is either a custom connector or a custom data export tool. Given the number and variation of financial software packages the unions use, some thought should be given as to the costs of each approach and how these costs will be borne by DOL and the individual unions.

A data export component that creates an intermediate file for subsequent import by the Data Preparation Subsystem is likely to be less complex and hence require less effort to develop than a fully integrated direct connector. Large unions have the capacity to hire programmers for creating programs that export data from their custom systems. Use of an intermediate file also makes more likely the possibility that a third-party extraction tool could be identified to perform all or part of the export processing required.

When determining who is responsible for developing the Data Import components, consider the following options:

DOL-supplied components. DOL develops the required connector technology for each of the primary financial packages. This option places the burden on DOL to provide customized applications for each of the selected financial packages. Additionally, this option would result in high costs for DOL and would carry a higher risk factor because of the number and variation of financial packages.

Third-party components. DOL contracts with third party vendor(s) to develop a program for each of the primary financial packages used by the union. This option carries a lower risk factor for success, but only after assuming that a third-party program will exist for each of the primary financial packages. However, third party programs for custom financial packages, as used by the larger unions, are unlikely.

Union components. Finally, DOL could simply supply just the base import OLFDS framework (the Connector infrastructure and a Standard File Connector). Each union would be responsible for developing or purchasing the appropriate export tool or developing a direct connector for its financial application. Unions bear most of the burden under this approach. They would be responsible for extracting data from their financial systems and formatting the data to the DOL standard. To ease this burden, DOL could provide a “translation” program that would work with the standard export formats of the most frequently used financial packages and thus share the burden with the unions.

These options are not mutually-exclusive. DOL may choose to provide “out of the box” connectors for only the most frequently-used financial applications, and require unions to bear the cost of building or buying connectors for others.

5.3 Forms Transmittal

Section 3.3 presents a number of possible approaches for transmitting a completed, signed form from the unions to DOL. Consider the following factors when evaluating the approaches:

Technology. All of the approaches described can be implemented using currently-available technology. The Standalone Web Application, Email-Based, and Integrated Direct Connection approaches use the existing distributed communications protocols of the underlying platform (e.g. J2EE, .Net, or CORBA). The Integrated, Non-Web Services approach also uses the platform communications capabilities, but requires the ability to integrate a client-side proxy component with the forms preparation application. All the preparation applications we have examined provide this capability.

The most advanced integration approach we propose is the combination of an integrated desktop component with a web-services submission interface. While web services are a relatively new interface protocol (compared to CORBA’s IIOP or Java’s RMI), they are based on long-standing protocols and standards (HTTP, TCP/IP, and XML) and industry has been quick to provide robust, usable technology to implement them. Microsoft’s .Net uses web service-based invocation as its default communication protocol, and provides powerful tools for wrapping web services interfaces around application code. While J2EE is only now adding web services to the official platform standard, numerous robust third-party products are already available to deploy J2EE-based services across the Internet, and comparable tools are emerging on a continual basis.

Performance. The significance of forms transmittal performance depends on which forms preparation design is selected, as this will in turn determine the available network capabilities. In the case of a client-side desktop application, forms will be sent across the Internet with low latencies and widely varying bandwidth – depending on the union’s specific Internet connections. In the case of a web-hosted server application, forms will be sent between components running within the DOL network, potentially even running on the same host, where bandwidth is high and latency is minimal.

Platform-specific protocols (IIOP, RMI, and DCOM) are typically highly-optimized for high-speed network access, with low latencies and high bandwidth. These protocols are not typically designed for Internet-wide communication, and impose overhead that may degrade performance to unacceptable levels. These protocols would be appropriate for interaction between server-based components of some other transmittal design, such as an email handler or web service proxy and other OLFDS subsystems.

In contrast, web services protocols are, by nature, intended for communication across the Internet. They rely on simple web-oriented protocols (TCP/IP and HTTP) and do not require the overhead associated with transaction propagation, security context, and other requirements of platform-proprietary protocols. While the use of XML for application-level protocol exchange (SOAP) adds some degree of verbosity, most web services are typically coarse-grained interfaces that pass as much data as possible with each call, further minimizing network round trips.

If a web-based forms application is used, the Integrated, Direct Connection approach (connecting a server-based forms preparation application with a forms submission handler running on the same

SOAP (Simple Object Access Protocol) provides a way for applications to communicate with each other over the Internet, independent of platform.

server) would offer the best overall performance overall because components would communicate using local function calls.

Finally, the Standalone Web Application approach, based on HTTP and HTML, shares many of the performance characteristics of the web-services approach. Since it is an interactive application, however, the need to transmit presentation-level data, such as HTML pages and HTML images, will require more bandwidth than a low-level web service, which would pass only data.

Interoperability. Front End choices also drive the degree to which client- and server-side components must support the same communications protocol. If a web-based application is used, it will most likely be built using the same server platform (J2EE, .Net, and others) as the rest of the Back End processing framework. In this situation, interoperability is not a major concern, as components can interact using whatever protocol the platform best supports.

Conversely, if the Data Preparation Subsystem and Back End submission are running on different hosts, they could be built using different technology (for example, a Windows-based COM component using a J2EE-based submission handler). In this situation, a web services-based interface would be particularly valuable. The strengths of the web as an information distributor – namely, the web’s simplicity of access and ubiquity – have been instrumental in resolving the fragmented middleware world where interoperability is hard to achieve. Web services complement these platforms by providing a uniform and widely accessible interface than services that are more efficiently implemented in a traditional middleware platform.

In the case of a standalone web application that transmits data using a web browser and HTTP, interoperability is not a significant issue. As long as the web application is built using web standards (HTTP and HTML), any existing browser can be used to transmit submitted forms and check processing status.

Security. All of the approaches at some point transmit information between filers and DOL, either as part of forms preparation or in subsequent submittal steps. Due to the open nature of the Internet, any time this information is sent it is at risk for disclosure or alteration by third parties. Since the filed reports are ultimately intended for public dissemination, confidentiality (and the associated use of techniques such as encryption) is not a significant issue. However, any communications channel must preserve the integrity and proof of authorship of submitted information, particularly since these documents are official government records, with associated document controls per NARA regulations.

In the cases in which a signed document is sent as a complete document via web-service, email, or as part of a browser application, digital signature technology should provide sufficient protection against tampering. If a third-party were to intercept and modify the information, the signer’s digital signatures would no longer match the data and the validation would fail. If the form is prepared online, a secure communications protocol, such as SSL, should be used to protect the interaction between users and DOL.

Integration. The Standalone Web Application and Email-Based approaches rely on the user to manually transfer the signed report form as an uploaded file or attachment, respectively. There is no direct integration with the Data Preparation Subsystem – all transfer is through the (shared) forms document. This approach makes for a simpler architecture, as no client-side components are required, and eliminates any dependency on the specific forms application. However, the decoupled aspect of this approach also requires more work from the user, making the submission process itself more complicated. Users must manually keep track of file locations, server

addresses, file numbers, and other detailed information because no client-side context is associated with the submission process.

In contrast, the various integrated interface approaches (whether based on web services, platform protocols, or direct connections) provide a simpler and more seamless user interface. Users can invoke the import functionality from within the forms application; they do not need to keep track of data file locations, server addresses, and other details. Because this approach requires more integration effort with the forms application, it requires a more complicated architecture and, consequently, more effort to develop.

5.4 Forms Processing

Section 4.1 identified two alternative models for the Processing Subsystem: (1) an immediate model, where submissions were validated and stored as soon as they were presented, and a (2) a deferred model, where submissions were first saved in temporary working storage and processed at some later time. Consider the following issues when evaluating the alternatives:

Complexity: The immediate processing model is simpler. All processing is performed within the context of a single service call, requiring minimal request-handling logic. Since no user state is retained across submissions, no intermediate storage or mechanisms for managing it is required. A simplified design requires a shorter development effort, with correspondingly lower costs, as well as reduced systems configuration and management costs. The user interface expressed through the Data Transmit Subsystem is also simplified – the result of a “submit” command is either “accepted” or “rejected” (with errors shown), providing relatively immediate feedback.

The deferred model requires an intermediate submission management facility, including persistent state management features to allow a filer to check submission status during the processing cycle. The subsystem interface and user interface are also made more complex as a result of this “two-part” submission approach. All this complexity will affect the development time of the system, as well as ongoing support and maintenance costs.

Performance and scalability. Because unions submit their financial reports at the end of their fiscal year, and because many unions follow the same fiscal calendar, the expected activity for the submission process will be highly variable. Many requests will arrive during a relatively limited number of days, and a negligible number of requests will arrive during most days of the year. If forms processing is handled immediately, there will be peak periods in which server load is very high. System response time will be severely degraded during peak periods. As a result, the “immediate” feedback of a submittal request will be greatly reduced. DOL could address this by adding more server capacity (extra memory, more CPUs, and greater bandwidth), but due to the variable nature of the load, much of this capacity will go largely unused during non-peak periods.

The deferred model separates the initial submittal request from the actual processing effort. This allows OLFDS to provide a more consistent response time, as submission is now a lightweight request. Processing can be performed during periods of light activity, such as non-business hours or weekends, smoothing out the bursts of activity. The overall effect is a more balanced use of server resources, allowing capacity to be efficiently allocated by DOL.

Business process support. The immediate approach implements a simplified business process, consisting of a few relatively simple validation steps and a final storage step, with validation errors terminating the workflow immediately. If additional steps are later added, the resulting workflow may not be amenable to such “straight-through” processing. In particular, if any of the

steps require some form of human review or intervention, the immediate model would not support this.

The deferred model can support a much more sophisticated processing flow. For example, in the current workflow model, any validation errors require the filer to resubmit the entire form. An extended processing system could allow OLMS staff to review a rejected form and override the automated validation step or correct errors manually; this would preserve changes through an audit mechanism. This corresponds more closely to the traditional paper-based review process.

5.5 Forms Storage

Assuming forms are represented using some form of XML-based document, such as XForms, XFDL, or XFA, several approaches to storing form content and associated data elements in a database are feasible.

Native XML Databases (NXD) excel at storing document-oriented data (XHTML or DocBook, for example), data that has a very complex structure with deep nesting, and data that is semi-structured in nature. Basically, if the data is represented as XML and is loosely organized, an NXD will be a good solution. An NXD can store any type of XML data, but may not be the appropriate tool to use for applications where the data is very well-defined and rigid. Because NXDs are a relatively young technology, tools, APIs, and other support is generally only available from the database vendor.

Conversely, a relational database is optimized for well-defined, repetitive structures which can be decomposed into a small set of relatively flat relations. Once the report data has been “shredded” into these tables, the resulting tables can be queried, transformed, and manipulated using highly-optimized, very mature technology. Because the LM forms are standardized and relatively simple, the strengths of a relational database seem to outweigh the additional costs of translating between the native XML form and the database table structure. Assuming a declarative mapping tool is used to drive this transformation, as opposed to programmatic code, any changes to form structure should be easily reflected in the storage process.

5.6 Web Query

The Query Subsystem presented in Section 4.3 is based on the traditional Model-View-Controller (MVC) design pattern commonly used in many web-based interactive systems. Its use can provide a scalable, decoupled design that separates data access, presentation, and application flow. As MVC is generally acknowledged as the most appropriate approach for this type of application, we have not presented any alternative design approaches. However, there are numerous approaches for taking the basic MVC framework and mapping it to specific implementation components. In particular, many of the application server vendors, including IBM, Oracle, and Microsoft, provide extensive web development frameworks that minimize the development effort required to create an MVC-based web application. These frameworks should be examined carefully in conjunction with available third-party reporting components when constructing the Query Subsystem.

5.7 Implementation Technology

Without a formal requirements analysis, a detailed design model cannot be fully defined. However, for each design model presented we have ensured that the required implementation technology exists. The degree to which current technology supports a particular option is a factor to be considered when weighing alternatives. The following sections provide a brief overview of

the critical implementation technologies that should be considered when defining the OLFDS solution.

5.7.1 Front End Application

Several COTS packages support electronic business forms with digital signatures. We examined most closely offerings from PureEdge and Adobe. Other vendors offer comparable products, including Silanis ApproveIt, Lexign ProSigner, and Evincible Secure Forms.

The descriptions below are meant to provide examples of available technology. They should not be interpreted as product recommendations or used as the basis for product selection.

5.7.1.1 PureEdge Internet Commerce System (ICS)

PureEdge (formerly UWI.Com) offers a set of tools that enables businesses to create, capture, process, and archive secure XML e-forms, and integrate them seamlessly with infrastructure and legacy systems with the explicit goal of supporting the creation and dissemination of legally-enforceable electronic documents. PureEdge clients include the Securities & Exchange Commission (SEC), JPMorgan Chase, and the U.S. Department of Defense.

PureEdge e-forms are based on Extensible Forms Definition Language (XFDL), an open XML protocol, as the native format throughout the e-form lifecycle. XFDL describes the e-form's complete template, business logic, and user-defined content. XFDL can contain supporting files, including graphics and attachments. Digital signatures are fully supported: each e-form can have multiple, overlapping signatures; signatures can be applied to all or part of the e-form, including other file attachments creating a "digital envelope." PureEdge e-forms integrate with Entrust and VeriSign PKI technologies as well as other electronic signature technologies.

The PureEdge ICS suite includes the following products:

- **ICS Viewer** is a main forms application that allows an end user to view, fill, and sign e-forms for secure, tamper-proof transactions. The viewer runs in any standard web browser or as a standalone application, with no server connection required. The component-based architecture allows enhanced functionality to be delivered easily through the creation of custom extensions, such as electronic signature support, or integration with other client-based applications, through the ICS API.
- **ICS Designer**, an application developers use to create e-forms, provides a drag-and-drop interface to layout forms and fields, define validation and signing logic, and establish document routing and workflow policies.
- **ICS API** is an e-Forms integration API library. IT provides a common, open interface to PureEdge e-forms that enables integration with client- and server-side applications including workflow, databases, and security infrastructures.
- **ICS Deployment Server** automates the installation and update of software, form templates, and other files for e-forms applications. It allows an application to centrally distribute and update software to reduce administrative overhead.
- **ICS Certificate Validation Module**, an ICS extension, integrates ACES applications with the MitreTek Certificate Arbitrator Module (CAM) for real-time validation of GSA-issued PKI certificates.

5.7.1.2 Adobe Accelio

Adobe's Accelio product family (formerly FormFlow) provides a comprehensive solution set for enabling data-centric business process automation, including a complete set of electronic forms products for data capture. The core of Accelio's e-forms solutions is XML Forms Architecture (XFA), a protocol for defining, validating and sharing document formats. XFA is an open, public specification that defines how a form will appear and act in an XML environment. XFA separates form data elements from the details of graphic presentations, allowing organizations to use XFA-based e-forms for a broad range of process management operations.

Adobe has thousands of Accelio product customers in more than 50 countries around the world, including 70 percent of the Fortune 100, 42 percent of the Global 500, and nearly every cabinet-level department of the U.S. Government.

Accelio provides the following business process solutions:

- **Accelio Capture** is a suite of products for designing and deploying electronic forms with built-in intelligence for a variety of delivery channels, including Web and handheld devices. Capture consists of a standalone e-forms application (Capture FormFlow), modules for web-based e-forms access (Capture Web, Capture Standard, and Capture Handheld), an e-forms development tool (Capture Designer), and a central server for managing and deploying software components and forms templates. End user modules support a rich set of features, including support for digital signatures and full Section 508 compliance.
- **Accelio Integrate**, an XML-based business process management solution, supports enterprise workflow and information. It allows organizations to integrate people, processes, and applications by automating existing business processes. Integrate includes a graphic process designer, a web-based user interface for work management, and various connectors for application integration.
- **Accelio Present** is a document output solution for creating personalized, customer-facing business documents from any data source. Documents can then be presented through a variety of delivery channels including the Web, wireless device, print, fax, e-mail and PDF. Present includes an output designer, the core output management engine, and various Output Paks for specific delivery channels and formats.

5.7.2 Back End Server Platform

Perhaps the most significant implementation choice for Back End subsystems is the software platform used for implementing components. The primary server-based distributed architectures are J2EE, Microsoft .Net, and CORBA.

5.7.2.1 J2EE

The Java2, Enterprise Edition (J2EE) platform provides models, technology, tools, and standards for building enterprise-scale, multi-tier applications. The J2EE environment features an open, Java-based component architecture that uses J2EE enterprise servers to host back-end components based on Java Server Pages (JSP) and web-based Java Beans. Businesses with more sophisticated processing needs can encapsulate their key business logic as powerful Enterprise JavaBeans, with a rich set of services and products providing transactions, security, resource management, and other application support.

J2EE provides an architectural model that is scalable and robust, while avoiding proprietary APIs and system platforms. As J2EE is an openly-available standard, there are a number of high-quality

commercial products that can be used, including BEA's WebLogic Server, IBM's WebSphere, and Oracle's 9iAS Application Server. These products are robust and well-engineered, and thus J2EE should be considered more mature than the comparable Microsoft .NET product. However, because J2EE tools support has not kept up with the runtime platforms, J2EE still requires more intensive programming and low-level integration effort.

5.7.2.2 Microsoft .NET

Microsoft's .NET provides a suite of developer tools, client applications, XML Web services, and servers necessary to create distributed solutions. .NET provides functionality and size of application comparable to J2EE's; however, .NET provides a more integrated and powerful development environment than those available for J2EE. A .NET solution can yield flexibility similar to a J2EE system, while providing superior integration when coupled with Microsoft databases and web-based applications. This reduces the burden on DOL and its development team because .NET provides a lot of desired functionality "out of the box." Because .NET is a "pre-fabricated" solution that makes development easier, it sacrifices the robust runtime features and scalability that J2EE-based products can offer. However, the .NET environment is less programming intensive than the J2EE environment and, consequently, would place less of a development burden on DOL.

5.7.2.3 CORBA

Another prevalent distributed component platform is the Common Object Request Broker Architecture (CORBA) architecture, as defined by the Object Management Group (OMG). Unlike web services-based systems such as .NET, CORBA components communicate using IIOP, a binary protocol optimized for inter-application communication. This environment requires the development of multiple CORBA-based systems that would interface through this standard protocol.

A CORBA-based system's primary advantage is that it enables communication between systems from any vendor, on almost any computer, operating system, programming language, and network. Although network-based, this type of architecture is not a lightweight web service application; consequently, CORBA-based products typically require robust hardware with which to operate such systems, varying from vendor to vendor. Because CORBA, like J2EE, is a complex multi-layer technology, implementation requires experienced developers.

5.7.3 Database

Another significant implementation decision is the choice of database product. Each database comes with its own implementation costs, speed, and security risks. SRA recommends that DOL weigh these factors as it determines the best combination for the application. Oracle's versatile database offers text indexing and an application server for controlling the database at the application level. DB2 and Informix databases are specifically suited for handling XML documents and related structures. This XML record schema for organizing the data will be intuitive to the system and require little translation between representative forms. However, if the .Net platform is used, Microsoft SQL Server provides out-of-box integration with the .Net infrastructure.

OMG is a consortium of over 700 companies. Its goal is to provide a common framework for developing applications using object-oriented programming techniques.

5.8 Operational Feasibility

Operational feasibility involves an assessment of the support needed to maintain and operate the baseline OLFDS enterprise. As defined in the baseline, the Front End requires minimal operational support. If a COTS product is selected as part of the Data Preparation Subsystem, the COTS vendor will provide support for distribution of the software (generally over the Web) with little or no user intervention. If a web based service is chosen as an alternative, once again the service will be self sustaining with minimal user support for system operation. However, if DOL proposes to supply custom software for data import from each financial package, it will have to build the capabilities for supporting a large user base and maintaining different versions of the software. From an operational perspective, it would be far simpler and more economical to DOL to publish a data standard and allow the unions to maintain the software required for transforming their data to the standard format.

At the Back End, most of the operations are self sustaining. The only user support that is required is the routine operation of servers, web site maintenance, and database backup, including database administration. NTIS has all the capabilities for supporting such a system. NTIS is currently supporting the LM-2 (current form before revision) electronic submission system in terms of data capture. NTIS also hosts the data for the public to query over the Internet. OLFDS is very similar to the current system that is operational. They have the requisite experience with different web servers and databases. OLMS should be able to support this OLFDS system in house, since it already has the capabilities of operating web servers and databases.

5.9 Technical Recommendation

The previous sections described the risks and benefits associated with each of the various OLFDS design alternatives. A detailed requirements gathering and analysis process should be performed before any definitive detailed design can be produced. However, based on our current understanding of OLMS business goals and technical requirements, SRA recommends the following approach to designing the OLFDS.

- COTS-based desktop forms application for creating and preparing union report forms
- Connector-based design for Data Import, supporting both export-file and direct-connection application integration, with DOL perhaps providing export tools for the most commonly used financial packages
- Integrated web-services transmittal service, using an integrated extension component to encapsulate communications protocols
- Server-based deferred-processing submission manager
- Relational database with XML extensions for storing original forms documents and extracted report data
- MVC-based web application, leveraging existing application frameworks and reporting components as much as possible
- J2EE or .Net for the underlying platform architecture

6 DOL Costs Estimate

This section presents the costs estimates for developing and deploying the recommended system (baseline system) and annual operational costs. [Section 8 addresses the burden on the labor unions in changing their chart of accounts to capture the additional financial information for the new LM-2s and a burden associated with the annual filings.] Section 6.1 provides estimated development (non-recurring) costs. Section 6.2 provides estimated annual operation and maintenance costs.

6.1 Non-recurring Costs for System Development and Deployment

The non-recurring costs for system development are calculated using cost categories and definitions listed in Table 6.1.

Table 6.1: Cost Element Structure and Definitions

| Cost Element | Definition |
|-------------------------------|---|
| System & Software Development | The labor costs to design, develop, and implement the system and software |
| Software Procurement | Costs to purchase COTS software for the system, including application software and software licenses |
| Hardware Procurement | Costs to purchase hardware for the system, including servers, routers, switches, data storage devices, other equipment, and system software |

6.1.1 Methodology

SRA followed a four-step methodology to estimate labor costs associated with system development:

- 1. Define the system to a subsystem level.** The technical solution recommended in Section 5.8 of this document is used as the basis for all non-recurring system development cost estimates. The baseline system is comprised of six subsystems: Import, Preparation, Transmittal, Processing, Storage, and Query.
- 2. Estimate labor rates.** Labor rates have a large amount of variation from contractor to contractor. To characterize a range of costs, labor rates were estimated by sampling GSA labor rate schedules for six companies. The companies that were selected met the following criteria:
 - Presence in the Washington DC metropolitan area
 - Develops software
 - GSA rate schedule available on-line
 - Striated sample required two companies being selected from 3 different size categories: Large companies (over \$1 billion in annual revenue); medium companies (between \$100 million and \$1 billion in annual revenue); and small companies (less that \$100 million in annual revenue)

Three values were calculated in the labor costs estimates:

- **Low rate** is the average of the two lowest rates for the given labor category
- **Median rate** is the average of the middle two rates for each given category
- **High rate** is the average of the highest two rates for each given category.

Each hourly labor rate was multiplied by 173.3 (hour per month using 2080 as a standard year) to produce the monthly labor rates. The labor rate used in subsequent labor costs estimates are given in Table 6.3.

Table 6.3: Average Labor Rates (in Dollars) per Labor Category

| Labor Category | Low Hourly Rate | Median Hourly Rate | High Hourly Rate | Low Monthly Rate | Median Monthly Rate | High Monthly Rate |
|--------------------------|-----------------|--------------------|------------------|------------------|---------------------|-------------------|
| System Architect | 85.22 | 138.78 | 210.00 | 15,966 | 23,035 | 34,026 |
| Senior Software Engineer | 69.10 | 105.85 | 148.00 | 13,410 | 18,035 | 24,502 |
| Software Engineer | 50.20 | 83.68 | 124.80 | 9,884 | 14,064 | 20,349 |

3. ***Estimate labor hours and labor categories to develop each subsystem.*** SRA based its estimate of effort required to develop and deploy the system on comparable system and software projects it has completed. Because of the complexities of the system, identifying a single labor category would not reflect actual costs to perform the work. Therefore, three labor categories – software engineer, senior software engineer, and software architect – were used to estimate the hours required to complete system development.

Estimates for person-months per labor category were made for each of the three development steps and then totaled. The development steps are:

- Requirements
- Development
- Test and Implementation

With the estimation of person-months of effort for the non-recurring development and monthly labor rates, labor costs for each labor category were calculated and are presented in Table 6.5. Additionally, a 25 percent burden was added to the monthly labor rates to cover project management, quality assurance, and administrative support. The burden estimate was based on historical contract norms provided by SRA's contract pricing department.

Table 6.5: Average Labor Rate and Estimated Development Costs

| Labor Category | Total Person-Months | Low Monthly Rate | Median Monthly Rate | High Monthly Rate | Low Estimated Labor Costs | Median Estimated Labor Costs | High Estimated Labor Costs |
|--------------------------|---------------------|------------------|---------------------|-------------------|---------------------------|------------------------------|----------------------------|
| System Architect | 19.8 | \$15,966 | \$23,035 | \$34,026 | \$395,152 | \$570,127 | \$842,148 |
| Senior Software Engineer | 53.5 | \$13,410 | \$18,035 | \$24,502 | \$896,119 | \$1,205,211 | \$1,637,373 |
| Software Engineer | 51.5 | \$9,884 | \$14,064 | \$20,349 | \$636,057 | \$905,036 | \$1,309,480 |
| TOTAL | | | | | \$1,927,328 | \$2,680,373 | \$3,789,001 |

The estimated labor cost for the development and deployment of the baseline system range from \$1,927,328, using the low labor rates, to \$3,789,001, using the high labor rates. Using median labor rates, the cost is \$2,680,373.

4. **Estimate hardware and software costs.** In addition to labor, deploying the recommended computing system will also require hardware and software procurements. Table 6.7 presents projected software and hardware based on the size and loading data of the technical architecture.

Table 6.7: Estimated Hardware and Software Costs

| Cost Element | Description | Estimated Costs |
|----------------------|--|-----------------|
| Software Procurement | Based on: DB2 database license (4 cpu), WebSphere database application server license (2 cpu), and PureEdge software form handling software. | \$325,638 |
| Hardware Procurement | Based on: Database server Dell PowerEdge 6650 (4 cpu), and a Web server Dell PowerEdge 2650 (2 cpu). | \$34,460 |

6.1.2 Estimated Non-recurring Costs

The total estimated costs, including labor, hardware, and software, for developing and deploying a baseline on-line financial downloading system are presented in Table 6.9. Because a range of cost estimates was desired, the hardware and software estimates were each multiplied by a 0.75 factor to get the low estimate and multiplied by 1.5 to get a high estimate. These values were then used with the low, median, and high labor values to result in total costs estimates given as Low, Medium, and High. All hardware and software procurement costs would be incurred the first year. It was assumed the baseline system would be located behind an existing firewall, so no cost was added for security.

Table 6.9: Total Development and Deployment Costs

| Cost Element | Low Estimate | Median Estimate | High Estimate |
|---------------------------------------|--------------|-----------------|---------------|
| System & Software Development (labor) | \$1,927,328 | \$2,680,373 | \$3,789,001 |
| Software Procurement | \$244,228 | \$325,638 | \$488,456 |
| Hardware Procurement | \$25,845 | \$34,460 | \$51,690 |
| TOTAL | \$2,197,401 | \$3,040,471 | \$4,329,147 |

6.2 Annual Operating Costs

The recurring annual costs for operating the baseline system over a three-year period are calculated using cost categories and definitions listed in Table 6.11.

Table 6.11: Cost Element Structure and Definitions (Annual Costs)

| Cost Element | Definition |
|------------------------|--|
| Personnel Support | The labor costs to maintain and operate the software and system |
| Software Maintenance | Costs for application software and system software annual maintenance fees |
| Broadband Connectivity | Costs for broadband Internet access and services |

- **Personnel support.** SRA followed a four-step methodology, similar to that used in Section 6.1 to estimate labor costs for personnel support:
 1. Define the labor tasks to operate and maintain the system (listed in Table 6.13)
 2. Estimate labor hours and labor categories for each task based on comparable projects
 3. Apply labor rates and calculate labor costs
 4. Total all costs for annual operating cost estimate

Table 6.13: Operations and Maintenance Tasks

| Task | Definition |
|-----------------------|---|
| System Administration | General computer operations and maintenance, help desk support |
| Software Maintenance | Software update installation, configuration management, bug fixes and patches |
| Database Maintenance | Database backups, archival, index analysis, performance tuning |
| Help Desk | User support and training |

Each of the tasks found in Table 6.13 was assigned a labor category, as presented in Table 6.15, and weekly estimates were made to accomplish each of the tasks. Additionally, yearly hours were calculated using a standard 2080-hour work year.

Table 6.15: Labor Estimates

| Labor Category | Hours per Week | | | Hours per Year | | |
|------------------------|----------------|--------|--------|----------------|--------|--------|
| | Year 1 | Year 2 | Year 3 | Year 1 | Year 2 | Year 3 |
| System Administrator | 16 | 16 | 16 | 832 | 832 | 832 |
| Systems Programmer | 4 | 8 | 12 | 208 | 416 | 624 |
| Database Administrator | 8 | 8 | 8 | 416 | 416 | 416 |
| Help Desk Staff | 20 | 20 | 20 | 1040 | 1040 | 1040 |

The labor rate structure used here was used in Section 6.1. However, because the estimate extends for three years, only the Median Labor Rates were used in calculating the labor estimate. The recurring annual labor costs were calculated by multiplying labor hours by labor rates for each of the labor categories and are presented in Table 6.17. An inflation factor of 4.1 percent was then applied to the present-day dollar estimate. The inflation factor was based on the average of the last three years' rate increases for government contracts.

Table 6.17: Labor Costs for Operation and Maintenance Tasks

| Labor Category | Labor Rate | Hours per Year | | | Estimated Labor Cost per Year | | |
|------------------------|------------|----------------|--------|--------|-------------------------------|-----------|-----------|
| | | Year 1 | Year 2 | Year 3 | Year 1 | Year 2 | Year 3 |
| System Administrator | \$83.68 | 832 | 832 | 832 | \$69,622 | \$69,622 | \$69,622 |
| Systems Programmer | \$105.85 | 208 | 416 | 624 | \$22,017 | \$44,034 | \$66,050 |
| Database Administrator | \$105.85 | 416 | 416 | 416 | \$44,034 | \$44,034 | \$44,034 |
| Help Desk Staff | \$51.50 | 1040 | 1040 | 1040 | \$53,560 | \$53,560 | \$53,560 |
| TOTAL | | 2496 | 2704 | 2912 | \$189,232 | \$211,249 | \$233,266 |
| TOTAL (w/inflation) | | | | | \$196,991 | \$238,312 | \$263,149 |

Total annual operation and maintenance cost estimates are presented in Table 6.19 and include personnel, software maintenance fee (the maintenance fee is calculated as 15 percent of the software license cost), and broadband connectivity fees. An inflation rate of 2.5 percent was used for software maintenance and broadband connectivity and based on actual FY02 rate increases.



Table 6.19: Annual Operations and Maintenance Costs Estimates (With Inflation)

| Cost Element | Year 1 | Year 2 | Year 3 |
|------------------------|---------------|---------------|---------------|
| Personnel Support | \$231,830 | \$276,067 | \$300,904 |
| Software Maintenance | \$48,846 | \$50,067 | \$51,318 |
| Broadband Connectivity | \$17,400 | \$17,835 | \$18,281 |
| TOTAL w/Inflation | \$298,076 | \$343,969 | \$370,504 |

7 Bookkeeping/Financial Accounting Software Survey

7.1 Objective

To validate the feasibility of the proposed On-Line System's architecture – specifically, the Front End Data Import Subsystem, SRA surveyed the bookkeeping/financial accounting packages that unions use to support their bookkeeping/financial accounting functions.

7.2 Methodology

DOL provided the source of survey items: a spreadsheet listing 20 software packages; unions using 14 of the packages; and the range of union size by receipts amount – for example, less than \$200,000, between \$200,000 and \$500,000, and greater than \$1 million – using the top nine packages. The list also indicates that Quickbooks, Quicken, and Peachtree are the three most commonly used accounting software packages.

The package types range from commercially available off-the-shelf solutions, such as Peachtree; to off-the-shelf, but not commercially available solutions, such as Client Bookkeeping Solution⁴; to custom, proprietary packages, such as Roberts Custom Software. The functionality of the listed products ranges from dues/receipts management, such as ASI, to the full financial accounting lifecycle, such as Quickbooks.

SRA conducted the software survey by collecting product information about the packages' data export/extraction functionality. Information sources included:

- Vendor web sites, including Quickbooks.com, Peachtree.com, and TMARresources.com (TIMSS), for information about COTS packages
- Union web sites to verify that the proprietary applications are used by the unions identified in the DOL source list
- Telephone contacts with COTS vendor customer support or product/business development staff, proprietary package customer support (help desk) staff, and union Information System staff

To identify a package's data extraction/export capabilities, the primary survey questions were:

- Is there an API or connection that allows an external application to extract transaction-level financial data from the application/system?
- Alternatively, can the application/system export data in structured file format (for example, comma- or tab-separated value, Excel, Access, or XML)?
- If the application/system does not support data export in a structure file format, can it generate a formatted report that can be saved to a file from which data can be extracted (for example, exporting a report to Excel, saving the report, and then extracting row/column information)?
- Does the application/system maintain data in a known storage format, such as a relational database, that can be queried separately if the data format were made available to DOL?

⁴ Creative Solutions does not sell Client Bookkeeping Solution (CBS) to the commercial market. The vendor sells its products only to accounting professionals. Accounting and auditing firms can either sell CBS to their clients or provide CBS to their clients as part of their services package.

- If the system does not support robust data export/extraction, what, if any, third-party applications support data export/extraction?

Table 1.1 lists the packages that were excluded from the survey:

Table 7.1: Packages Excluded From Survey

| Package | Reason(s) For Exclusion |
|---|---|
| "Unknown custom proprietary software" (used by SAG) | <ul style="list-style-type: none"> ▪ Not possible to identify for which system it was necessary to collect information |
| Titan | <ul style="list-style-type: none"> ▪ Unable to obtain information from IBT, which indicated that most locals have switched to Quickbooks |
| Real World | <ul style="list-style-type: none"> ▪ Vendor, Essential Integrated Data, Ltd., indicated that the product has been discontinued, and they no longer support the product |
| ULTRA | <ul style="list-style-type: none"> ▪ Unable to obtain information from Carpenters ▪ Automated Business Design offers an "Ultra Accounting" solution, but indicated that the Carpenters are not a client |
| ASI | <ul style="list-style-type: none"> ▪ LIUNA did not respond to request for information |
| Compass | <ul style="list-style-type: none"> ▪ IAMAW indicated that the product's functional scope is limited to tracking dues and fees collected from members – primarily to determine per-capita tax that lodges owe the international |
| LIBRA | <ul style="list-style-type: none"> ▪ LIUNA did not respond to request for information |
| WinStabs | <ul style="list-style-type: none"> ▪ Unable to obtain information from UTU |
| MS (Microsoft) Money | <ul style="list-style-type: none"> ▪ Target customer: Home user ▪ An organization using this product would likely continue to enter data manually into an on-line form |
| Money Counts | <ul style="list-style-type: none"> ▪ Vendor indicated that it is "pulling product off the shelf" ▪ Target customer: Home user ▪ An organization using this product would likely continue to enter data manually into an on-line form |

7.3 Survey Results

This section presents summary information for 10 of the 20 packages on the DOL source list.

Quickbooks

| | |
|-----------------|--|
| Vendor | Intuit |
| Type | COTS package providing full accounting lifecycle functionality. Customer target: small- to mid-size organizations. |
| Database | Not identified |

**Export
Functionality**

- Uses QBXML API
- Product limited to exporting lists, such as customer lists and chart of accounts
- Extensive standard reports and ability to define custom detail transaction reports that can be exported and saved as Excel files or as ASCII text, tab-delimited, or comma-delimited files
- Several third-party vendors offer products that enhance reporting and data export/extraction, including:
 - Peter's Software Data Flow Manager supports bi-directional data integration (import and export) between QuickBooks and any other data format
 - Datablox Office Q and Office Q Pro enable users to extract all accounting information from Quickbooks. It supports all current versions of Quickbooks. It enables data to be extracted to a number of applications (therefore, formats), including Word, Excel, Access, and SQLServer

Roberts Custom Software

| | |
|---------------------------------|--|
| Vendor | Roberts Custom Software |
| Type | Custom proprietary. Supports core financial accounting functions, including General Ledger, Accounts Receivable, Accounts Payable, and Fixed Assets. Can automatically download LM-2 Schedules 9 and 10. Standard reports provide LM-2 line information and totals; users can print reports and (manually) transfer data to the LM-2 form. |
| Database | Indexed file structure using RM-Cobol and Visual Basic |
| Export Functionality | <ul style="list-style-type: none"> ▪ Comma-delimited files ▪ Flat sequential files (preferred) |

Quicken

| | |
|---------------------------------|--|
| Vendor | Intuit |
| Type | COTS personal finance management solution. Customer target: Home user, very small organizations. |
| Database | Not identified |
| Export Functionality | <ul style="list-style-type: none"> ▪ Export capability using QIF (Quicken Interchange Format) files |



Peachtree

| | |
|-----------------------------|---|
| Vendor | Peachtree |
| Type | COTS package providing full accounting lifecycle functionality. Customer target: small- to mid-size organizations. |
| Database | Btrieve |
| Export Functionality | <ul style="list-style-type: none">▪ Supports exporting lists and a limited set of accounting data as comma-separated value (.csv) files▪ Extensive standard reports and filtering capabilities; reports can be saved as Excel files.▪ Several third-party vendors offer products that enhance reporting and data export/extraction, including:<ul style="list-style-type: none">▪ Crystal Decision's Crystal Reports for Peachtree expands the standard reporting capabilities by provides unlimited access to data for reporting Multitware's PawCOM provides access to all of the underlying tables in Peachtree. An ActiveX control, PawCOM offers both read and write access to Peachtree, and is compatible with Peachtree Accounting and Complete Accounting for Windows version 7 or later |

KI

| | |
|-----------------------------|---|
| Vendor | KI Technology |
| Type | Custom accounting software featuring General Ledger, Accounts Payable, Payroll, and Accounts Receivable modules. Approved by UAW International to provide accounting solutions to its locals. Supports mapping LM accounts to LM-2 form, with the ability for user override. This past year, KI added LM-2 Schedules 9 and 10 support by creating an interface from the payroll system. |
| Database | Not identified |
| Export Functionality | <ul style="list-style-type: none">▪ Exports data in ASCII comma-delimited or fixed field format▪ Does not support export to special formats, such as Excel▪ Does not currently support transaction detail export. However, logic exists; therefore, functionality could be available by February 2003 (for UAW's next LM-2 reporting cycle) |



Solomon

| | |
|-----------------------------|---|
| Vendor | Great Plains Software |
| Type | COTS package providing full accounting lifecycle functionality. Customer target: small- to mid-size organizations. |
| Database | Version dependent |
| Export Functionality | <ul style="list-style-type: none">▪ API available through SQL-type interface▪ Reports can be pulled with backend reporting tools, such as Crystal Data or FRX, which is integrated into the tool |

Client Bookkeeping Solution

| | |
|-----------------------------|---|
| Vendor | Creative Solutions, Inc. |
| Type | COTS package providing full accounting lifecycle functionality. |
| Database | Microsoft Access |
| Export Functionality | <ul style="list-style-type: none">▪ No API▪ File/table structures are protected and not documented. Though a user organization can access the underlying Access database, changes to structure invalidates support contract▪ Uses Crystal Reports report engine. Only canned reports are available; does not support user-defined reports▪ Exports data in a variety of formats, including ASCII-delimited, tab-delimited, ODBC (for Access and FoxPro), comma-separated value (.csv), Data Interchange File, Portable Data File (.pdf), Rich Text Format (.rtf), Excel, Word, HTML, and XML |

ICS

| | |
|-----------------------------|--|
| Vendor | N/A |
| Type | Custom proprietary. Tracks all receipts, but primary function is dues tracking. |
| Database | Paradox |
| Export Functionality | <ul style="list-style-type: none">▪ No API▪ Uses Shizam's Report Wizard, which allows data filtering▪ Reports can be exported as ASCII files▪ Supports exporting data to Paradox database |

LUMS (Labor Union Management System)

| | |
|-----------------------------|---|
| Vendor | System Design Associates (SDA) |
| Type | Custom proprietary. Functionality focuses on member information, including demographics and dues processing. |
| Database | Paradox |
| Export Functionality | <ul style="list-style-type: none">▪ "If there is a need to export data, [SDA] can do it"▪ Supports sending password encrypted .zip files and secure http and ftp transfers |

TIMSS (TMA Resources Integrated Members Services Solution)

| | |
|-----------------------------|--|
| Vendor | TMA Resources |
| Type | Custom proprietary. TIMSS is an Accounts Receivable module used primarily to track dues status and track membership information that locals enter. |
| Database | Oracle 8.5 and 8i |
| Export Functionality | Not identified |

7.4 Survey Analysis

Survey results indicate that most financial packages used by the unions support data export – either directly through the software or, in some cases, through third party integrated products that are supplied with the software. The financial packages used by unions can be broadly divided into COTS products and custom developed software. Most COTS products, other than Quicken, have export capabilities and also support third party reporting tools that can be used for exporting the data. Quicken is used by smaller unions with a small number of transactions. It is likely that Quicken users would choose to cut-and-paste or enter data directly into the LM-2 form. An alternative solution has been outlined in Section 1.1.1.

Most custom developed software applications have export capabilities for exporting the data and also use standard databases for storing data. Standard databases support SQL or other API for direct access to the data. This allows three options for Data Import into form LM-2. Using the export capabilities, the transaction data can be exported to a text file, and a custom program could convert data into a form acceptable to the Import Manager (see Section 1.1) for mapping to LM-2 fields. The second option would be to develop software that would access the tables directly and produce a file in the standard data interchange format. The third option would be to write software that will allow direct connection from the Import Manager for seamless transfer of data.

8 Preliminary Estimate of Burden Change on Filers

8.1 Summary

This section presents the preliminary estimate of burden on filers that would result from changes DOL proposes for Form LM-2 and proposed Form T-1. The estimates are based on the assumptions and methodology presented in Section 8.2. Table 8.1 summarizes the preliminary estimate.

Table 8.1: Summary of Preliminary Estimate of Burden on Filers

| Proposed Form | Estimated Non-Recurring Burden Per Filer | Estimated Incremental Annual Recurring Burden Per Filer |
|---------------------|--|---|
| Revised LM-2 | | |
| Burden Hours | 72.06 hours | 21.18 hours |
| Burden Cost | \$ 3,633 | \$ 511 |
| Proposed T-1 | | |
| Burden Hours | 3.46 hours | 3.31 hours |
| Burden Cost | \$ 96 | \$ 91 |
| Total | | |
| Burden Hours | 75.52 hours | 24.49 hours |
| Burden Cost | \$ 3,729 | \$ 602 |

8.2 Assumptions, Methodology, and Approach

8.2.1 Overall Assumptions

The burden change was estimated under the following six overall assumptions:

1. Because of the large differences between the smallest and largest LM-2 filers (based on annual reported receipts), the preliminary estimate of burden change was developed for different levels of LM-2 filers. Each level, or tier, has a different average burden estimate due to the differences in the scale of the organizations. These tiers were established based on the weighted average of annual receipts reported to DOL by each LM-2 filer in 2000. For this preliminary burden analysis, we have established three tiers of LM-2 filers:

- **Tier 1 Filers**

- LM-2 filers reporting annual receipts between \$200,000 and \$500,000.
- There were 2,260 Tier 1 filers reporting in 2000.
- The weighted average receipts per Tier 1 filer is \$345,000.

- **Tier 2 Filers**

- LM-2 filers reporting annual receipts between \$500,000 and \$50 million.

- There were 3,111 Tier 2 filers reporting in 2000.
 - The weighted average receipts per Tier 2 filer is \$3.263 million.
 - **Tier 3 Filers**
 - LM-2 filers reporting annual receipts greater than \$50 million.
 - There were 43 Tier 3 filers reporting in 2000.
 - The weighted average receipts per Tier 3 filer is \$193.0 million
2. The revised LM-2 will be submitted via the OLFDS Front End approach outlined in Section 3. Consequently:
- **Tier 1 Filers** are those who will either utilize a manual bookkeeping approach or a Commercial-Off-The-Shelf (COTS) accounting package and will utilize a cut-and-paste method for manually entering information in the LM-2. Tier 1 filers will not use an electronic data export approach because it is more cost-effective or convenient to continue the form-filling approach.
 - **Tier 2 Filers** are those who will utilize a COTS accounting package and will rely on a translation software application developed by the Department of Labor to export data in a text/XML format. Tier 2 filers will not use a form-filling approach for the bulk of their inputs because, given the organization size, it will not be cost-effective.
 - **Tier 3 Filers** are those who will utilize a customized accounting package and will develop a customized translation software application to export data in a text/XML format. Tier 3 filers will also not use a form-filling approach for the bulk of their inputs because it will not be cost-effective.
3. All filers will benefit from LM-2 and T-1 preparation efficiency improvements as they gain more submittal experience with the revised LM-2 and the proposed Form T-1 in out-years. Filing the LM-2 utilizing the Data Import Subsystem approach will also immediately reduce the burden on Tier 2 and 3 filers when they prepare the LM-2 Schedules and line items that do not change from the current Form LM-2.
4. LM-2 filers are responsible for filing the new Form T-1, Trust Annual Report. The estimated number of annual T-1 filings is based on a weighted average of estimated annual T-1 submittals for each LM-2 Tier group.
5. The preliminary estimate of burden change on filers assumes that the data reporting requirements proposed for the revised LM-2 and the proposed Form T-1 are already captured as a normal part of the filer's business practice.
6. This burden analysis is a preliminary estimate of the change in burden for the proposed LM-2 changes and the proposed Form T-1. The analysis does not include direct input from filers through interviews or surveys, nor has this analysis taken into consideration comments on the proposed changes in reporting requirements LM-2 filers may have submitted to DOL. These are steps, among others, that DOL ought to consider when submitting the burden estimates for these forms to the Office of Management and Budget.

8.2.2 Overall Methodology

Analysis Period. The preliminary burden change analysis is estimated over a three year period. Year 1 of the analysis includes non-recurring and recurring incremental burden. Years 2 and 3 include only recurring incremental burden. Provisions are made for LM-2 and Form T-1 preparation efficiency improvements during these out-years.

Applicable OLMS Forms: The burden analysis covers two DOL proposals:

- Changes to the current LM-2 burden for the proposed revisions to the LM-2 form and submission approaches.
- Development of a new preliminary burden estimate for the proposed Form T-1.

8.2.2.1 LM-2 Changes Methodology Summary

The preliminary estimate of the LM-2 burden change is based on a weighted average of changed burden estimates developed for the average filer in each LM-2 tier. The weighting is based on the number of LM-2 filers for each tier. Separate burden estimates are prepared for the non-recurring burden required to implement the proposed changes and for the recurring incremental burden of preparing and submitting the revised LM-2. The LM-2 burden change estimate is segregated into five elements:

- LM-2 preparation time
- LM-2 Pages 1 and 2 and Statements A and B
- LM-2 Schedules with no revisions
- LM-2 Schedules with revisions
- New LM-2 Schedules

The LM-2 burden change analysis relies on time impacts estimated for discrete activities anticipated to be performed in implementing the proposed LM-2 changes and preparing the proposed LM-2 form on an annual basis. The calculated burden hours for the proposed LM-2 changes are applied against historical filer labor hour rates reported to DOL for the filer labor categories expected to incur labor hours implementing the changes and preparing the proposed LM-2 form. The non-recurring and recurring burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

8.2.2.2 Form T-1 Methodology Summary

The preliminary estimate of the Form T-1 burden is based on a weighted average of burden estimates developed for the average filer in each LM-2 tier. The weighting is based on the total number of Form T-1s estimated to be filed annually for each tier. Separate burden estimates are prepared for the non-recurring burden required to implement the reporting requirements for the proposed Form T-1 and for the recurring incremental burden of preparing the submittal. The Form T-1 burden estimate is segregated into three elements:

- Form T-1 preparation time
- Form T-1, Questions 1 - 24
- Form T-1, Schedules 1 - 4

The Form T-1 burden analysis relies on time impacts estimated for discrete activities anticipated to be performed in implementing the proposed Form T-1 reporting requirements and preparing the T-1 submittal on an annual basis. The calculated burden hours for the proposed Form T-1 are applied against historical labor hour rates reported to DOL for the filer labor categories expected to incur labor hours implementing the proposed reporting requirements and preparing the proposed Form T-1 submittal. The non-recurring and recurring burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

8.3 Analysis – Summary of Proposed LM-2 Changes

This analysis compares the proposed revised Form LM-2 to the current Form LM-2. The analysis details the proposed changes to the LM-2 form by each discrete line item for Pages 1 and 2, Statements A and B, and Schedules 1 - 22. Changes are denoted as the following:

- Additional line-item requirements
- Revised line-item requirements
- Deleted line-item requirements
- No changes to line-item requirements

8.4 Analysis: Non-recurring Incremental Burden on Filers

8.4.1 Non-Recurring Burden Assumptions

1. Non-recurring costs are one-time costs incurred by the filer to implement:
 - The proposed LM-2 reporting changes and the new electronic reporting requirements
 - The reporting process for the proposed Form T-1
 - These costs will be incurred in Year 1 of the burden analysis
2. Non-recurring costs associated with the proposed changes to the LM-2 and the proposed Form T-1 fall into two categories: (1) non-recurring burden for software changes, and (2) non-recurring incremental accounting burden.

- **Non-recurring burden for software changes**

- Implementation of a form-filling software program developed by the Department of Labor.

Tier 1 filers will utilize the form-filling cut-and-paste method for a manual submission of the LM-2 and Form T-1. Tier 1 filers will utilize a form-filling software program provided by the Department of Labor to cut and paste or manually type in the reporting data in the LM-2 form and the Form T-1.

- Implementation of a translator software application

The burden analysis assumes that the revised LM-2 and the proposed Form T-1 will be submitted under the Front End Data Input Subsystem approaches outlined in the discussion of the On-Line Financial Downloading Concept. Tier 2 and 3 filers will utilize a translator software application to export LM-2 and Form T-1 reporting data in a text/XML format. Tier 2 filers utilize a COTS accounting package and will integrate a translator software application developed by DOL to export the reporting data. Tier 3 filers utilize a customized accounting package and will develop a customized translator software application to export the reporting data.

- **Non-Recurring Incremental Accounting Burden**

- Development of changes in the filer's account structure

All filers will adjust their chart of account structure as required to facilitate the export of reporting data.

- Development of changes in report query parameters

All filers will incur a non-recurring burden to implement changes to their existing data reports that are generated for preparation of the LM-2 form. All filers will also incur a non-recurring burden to develop additional data reports that will be required for the proposed additional LM-2 reporting requirements and the proposed Form T-1 reporting.

- Training of personnel

Personnel, including officers and employees, will need to learn about the proposed changes to the LM-2 and proposed Form T-1 reporting requirements.

- Reviews

Performance of management reviews of the accounting changes implementation process.

8.4.2 LM-2 Non-Recurring Burden Methodology

The estimate of the LM-2 non-recurring incremental burden is based on a weighted average of non-recurring burden estimates developed for each Tier. The weighting is based on the number of LM-2 filers for each Tier. The LM-2 non-recurring incremental burden estimate relies on time impacts estimated for discrete activities anticipated to be performed in implementing the proposed LM-2 changes. The calculated burden hours for the proposed LM-2 changes are applied against historical filer labor hour rates as reported to DOL categories expected to incur labor hours in implementing the proposed changes to the LM-2 form. The non-recurring burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

The LM-2 non-recurring incremental burden estimate is segregated into five elements:

1. Preparation time

- Installation, testing and management review of the LM-2 form-filling software provided by the DOL that will be utilized by all filers.
- Development or acquisition, testing and reviewing of the Front End data submission transfer software application to the filer's accounting systems. Tier 2 filers will install, test and review the translator software program developed by the DOL. Tier 3 filers will develop, test and review a customized translator software application.
- Acquisition costs for digital signatures for all filers.

2. LM-2 Pages 1 and 2 and Statements A and B

- Based on the proposed LM-2 changes, no non-recurring burden is expected to be incurred by LM-2 filers.

3. LM-2 Schedules with no revisions

- Based on the proposed LM-2 changes, no non-recurring burden is expected to be incurred by LM-2 filers.

4. LM-2 Schedules with revisions non-recurring discrete activities

- Schedule 5 – Investments

LM-2 filers are anticipated to incur a non-recurring burden for the following activities related to reporting requirements revisions for this schedule:

- Design of the revised data query report
- Development of the revised data report query parameters
- Testing of the revised data query report
- Management review of the revised data query report
- Documenting the revised query process
- Schedule 11 and 12 – Disbursements to Officers and Employees

LM-2 filers are anticipated to incur a non-recurring burden for the implementation of training for officers and reporting employees on the proposed time estimating requirements.

5. New LM-2 Schedules

- Schedules 1, 8, 13 - 22

LM-2 filers are anticipated to incur a non-recurring burden for the following activities related to the additional reporting requirements for these schedules:

- Implementing any changes in the chart of accounts
- Design of the additional data query reports
- Development of the additional data report query parameters
- Testing of the additional data query reports
- Management review of the additional data query reports
- Documenting the additional query processes
- Training of officers and employees on the proposed Schedule requirements

8.4.3 Form T-1 Non-Recurring Burden Methodology

The estimate of the Form T-1 non-recurring burden is based on a weighted average of burden estimates developed for each tier. The weighting is based on the total number of Form T-1's estimated to be filed for each tier. The Form T-1 burden change analysis relies on time impacts estimated for discrete activities anticipated to be performed in implementing the proposed Form T-1 reporting requirements. The calculated burden hours for the proposed Form T-1 are applied against historical filer labor hour rates as reported to DOL for the labor categories expected to incur labor hours in implementing the proposed Form T-1 reporting requirements. The non-recurring burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

The Form T-1 non-recurring burden estimate is segregated into three elements:

1. Preparation time

- Installation, testing and management review of the Form T-1 filling software provided by DOL that will be utilized by Tier 1 filers only.
- Tier 2 and 3 filers will utilize the translator software application implemented for the LM-2 and it is anticipated that they will incur no additional non-recurring burden for the proposed Form T-1.

2. Form T-1, Questions 1 - 24

- Based on the proposed Form T-1 reporting requirements for these questions, no non-recurring burden is expected to be incurred by Form T-1 filers.

3. Form T-1, Schedules 1 - 4

- Form T-1 filers are anticipated to incur a non-recurring burden for the following activities related to reporting requirements revisions for these Schedules:
 - Implementing any changes in the chart of accounts
 - Design of the additional data query reports
 - Development of the additional data report query parameters
 - Testing of the additional data query reports
 - Management review of the additional data query reports
 - Documenting the additional query processes
 - Training of officers and employees on the proposed Schedule requirements

8.5 Analysis: Recurring Incremental Burden on Filers

8.5.1 Recurring Burden Assumptions

1. Recurring costs are the additional ongoing costs incurred annually by the filer for:
 - The changed preparation and submittal requirements for the proposed LM-2 reporting changes and the new electronic reporting requirements.
 - The additional reporting requirements for the proposed Form T-1.
2. All filers will benefit from LM-2 and Form T-1 preparation efficiency improvements as they gain more submittal experience with the revised LM-2 and the proposed Form T-1 in out-years. Filing the LM-2 utilizing the Front End Data Input Subsystem approach will also immediately reduce the recurring burden on Tier 2 and 3 filers for preparation of the LM-2 Schedules and line-items that do not change from the current LM-2 version.
3. The preliminary estimate of the burden change on filers assumes that the data reporting requirements proposed for the revised LM-2 and the proposed Form T-1 are already captured as a normal part of the filer's business practice.
4. Recurring costs associated with the proposed changes to the LM-2 and the proposed Form T-1 include the following:
 - Recurring burden for the preparation of additional query reports and data export files for the proposed LM-2 changes and the Form T-1.
 - Recurring burden for the additional form-filling cut and paste tasks for Tier 1 filers only.
 - Recurring burden for the additional management time to review the LM-2 and Form T-1 submittals.

8.5.2 LM-2 Recurring Incremental Burden Methodology

The preliminary estimate of the LM-2 recurring incremental burden is based on a weighted average of recurring burden estimates developed for each tier. The weighting is based on the number of LM-2 filers for each Tier. The LM-2 recurring incremental burden estimate relies on time impacts estimated for discrete activities anticipated to be performed in preparing the proposed LM-2 submittal. The calculated burden hours for the proposed LM-2 changes are applied

against historical filer labor hour rates as reported to the Department of Labor for the labor categories expected to incur labor hours in preparing the proposed LM-2 submittal. The recurring incremental burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

The LM-2 recurring incremental burden estimate is segregated into five elements:

1. Preparation time

- The additional recurring burden for review of the proposed LM-2 form and filing instructions and for management review of the proposed LM-2 submittal.

2. LM-2 Pages 1 and 2 and Statements A and B

- The average burden per line-item for this LM-2 element is based on the OMB-approved LM-3 burden minutes. Pages 1, 2 and Statement's A, B and the Officer Disbursements Schedule are common to both the current LM-2 and the current LM-3. The LM-3 contains a total of 58 line-items consisting of Pages 1, 2 and Statement's A, B and the Officer Disbursements Schedule. The OMB-approved recurring burden for the LM-3 is 382.20 minutes for the 58 LM-3 line items which calculates to 6.59 average minutes per LM-3 line item. The proposed LM-2 has 74 line-items for these same parts of the form which is an increase of 16 additional line items. The incremental recurring burden time is calculated by applying the average 6.59 minutes per line-item to the additional 16 line-items for the proposed LM-2.

3. LM-2 Schedules with no revisions

- The proposed LM-2 Schedules that have no changes are Schedule's 2, 3, 4, 6, 7, 9 and 10.
- The current average recurring burden for each of these Schedules is 35.29 minutes which is calculated in the following manner. The current LM-2 approved recurring burden estimate is 876.2 minutes. The current LM-3 approved recurring burden estimate is 382.2 minutes. Subtracting the LM-3 burden from the LM-2 burden results in an additional time to complete the current LM-2 of 494.0 minutes. This additional burden is for the 14 additional schedules in the current LM-2 versus current LM-3. The average burden per current LM-2 Schedule of 35.29 minutes is calculated by dividing the additional time of 494.0 minutes to complete the current LM-2 versus the current LM-3 by the 14 additional current LM-2 Schedules.
- The recurring burden change for these Schedules is a reduction in burden of 50 percent from the current LM-2 average of 35.29 minutes due to the utilization of the electronic data export capability for Tier 2 and 3 filers. Tier 1 filers will not have any recurring burden reduction since they will not utilize the proposed electronic data export approach for the LM-2 submittal.

4. LM-2 Schedules with revisions

- Schedule 5 – Investments

The recurring burden change for this Schedule is a reduction in burden of 50 percent from the current LM-2 average of 35.29 minutes due to the utilization of the electronic data export capability for Tier 2 and 3 filers. Tier 1 filers will not have any recurring burden reduction since they will not utilize the proposed electronic data export approach for the LM-2 submittal.

- Schedule 11 and 12 –Disbursements to Officers and Employees

LM-2 filers are anticipated to incur a recurring incremental burden for the following activities related to reporting requirements revisions for these schedules:

- Preparation of time estimates.

The recurring incremental burden is calculated based on reporting officer and employee counts for an average filer in each Tier.

- Preparation of consolidated time estimate reports.
- Data cut and paste operations for Tier 1 filers only.
- Preparation of the data export files for Tier 2 and 3 filers.
- Execution of the data reporting edit/validate/transmit operations for Tier 2 and 3 filers.
- An efficiency improvement factor is applied to the Year 1 burden calculations for Year 2 and 3 in the analysis.

5. New LM-2 Schedules

- Schedule's 1, 8, 13 - 22

LM-2 filers are anticipated to incur recurring incremental burden for the following activities related to the additional reporting requirements for these Schedules:

- Preparation of data query reports and cut and paste operations for Tier 1 filers. For Schedules 14 – 22, the recurring burden is calculated based on the average annual volume of receipt and disbursement transactions estimated for a Tier 1 filer.
- Preparation of the data export files for Tier 2 and 3 filers.
- Execution of data reporting edit/validate/transmit operations for Tier 2 and 3 filers.
- An efficiency improvement factor is applied to the Year 1 burden calculations for Year 2 and 3 in the analysis.

8.5.3 Form T-1 Recurring Incremental Burden Methodology

The estimate of the Form T-1 recurring burden is based on a weighted average of burden estimates developed for each tier. The weighting is based on the total number of Form T-1's estimated to be filed annually for each tier. The Form T-1 burden analysis relies on time impacts estimated for discrete activities anticipated to be performed in preparing the proposed Form T-1 submittal. The calculated burden hours for the proposed Form T-1 are applied against historical filer labor hour rates as reported to DOL for the labor categories expected to incur labor hours in preparing the proposed Form T-1 submittal. The recurring burdens are applied against a blended average labor rate weighted by the estimated burden percentage for each labor category.

The Form T-1 recurring burden estimate is segregated into three elements:

1. Preparation time

- The recurring burden for review of the proposed Form T-1 form and filing instructions and for management review of the proposed Form T-1 submittal.

2. Form T-1 Questions, 1 - 24



- The recurring burden time is determined by applying the average 6.59 minutes per line-item calculated for the same line-items on the LM-2 to the line-items for Form T-1 Questions 1 – 24.

3. Form T-1 Schedules, 1 - 4

- Schedule 1 - 3

Form T-1 filers are anticipated to incur recurring burden for the following activities related to the additional reporting requirements for these Schedules:

- Preparation of data query reports and cut and paste operations for Tier 1 filers. For Schedules 1 and 2, the recurring burden is calculated based on the average annual volume of receipt and disbursement transactions estimated for a Tier 1 filer.
- Preparation of the data export files for Tier 2 and 3 filers.
- Execution of data reporting edit/validate/transmit operations for Tier 2 and 3 filers.
- An efficiency improvement factor is applied to the Year 1 burden calculations for Year 2 and 3 in the analysis.
- Schedule 4
 - The recurring burden is based on the average recurring burden for LM-2 schedules that do not change of 35.29 minutes. This schedule is the same as the LM-2 Schedule 2 which does not change under the proposed LM-2 revisions.
 - The calculation of recurring burden for this Schedule is a reduction in burden of 50% from the current LM-2 Schedule 2 average of 35.29 minutes due to the utilization of the electronic data export capability for Tier 2 and 3 filers. Tier 1 filers will not have any recurring burden reduction since they will not utilize the proposed electronic data export approach for the Proposed Form T-1 submittal.